MATRIX METHODS IN STRUCTURAL DYNAMICS

by

Charan Phimphilai

Submitted in Partial Fulfillment of the Requirements

for the Degree of

Master of Science

in the

Civil Engineering

Program

Advisor — Date

Dean of the Graduate School — Date

YOUNGSTOWN STATE UNIVERSITY

January, 1978

ABSTRACT

MATRIX METHODS IN STRUCTURAL DYNAMICS

Charan Phimphilai

Master of Science

Youngstown State University

The purpose of this thesis is to formulate a usable set of computer programs written in FORTRAN IV computer language which are associated with problems that arise in the field of matrix operations in structural dynamics. Each program is accompanied by a review of the matrix theory, a complete flow chart, a print out of the actual computer program, and a sample example to illustrate the results.

This work is divided into two distinct parts. The first section includes a series of programs which analyse the determinant problem, the matrix inversion problem, the characteristic value problem, the characteristic vector problem, the normalization problem and the Cholesky Triangularization (1)* method. In the second section a computer program using finite

* number in parenthesis refers to literature cited in the bibliography

difference techniques is written to determine the dynamic response of a lumped-mass structural system subject to external time-varying loading conditions.

The advantage of the latter method of analysis is that it completely eliminates the classical approach to the solution of the problem which includes both the necessity of computing the natural frequencies and modal shapes of the free vibration problem, and the utilization of a series-type integral solution for the problem.

ACKNOWLEDGEMENTS

The author wishes to acknowledge his deep appreciation and gratitude to Dr. Paul X. Bellini, his thesis advisor, whose guidance, efforts, time and encouragement directly contributed in the completion of this thesis.

The author also wishes to thank his review committee, Dr. Michael K. Householder and Professor John F. Ritter for giving their valuable time toward the completion of the requirements of his work.

Great appreciation is given to my parents Mr. & Mrs. C. Phimphilai for supporting my studies.

TABLE OF CONTENTS

LIST OF SYMBOLS

SYMBOLS	DEFINITION
$[A]$	Symmetric matrix
$[F]$	Force matrix
$[I]$	Identity matrix
$[K]$	Elastic bending stiffness matrix
$[L]$	Lower triangular matrix
$[M]$	Mass matrix
$[U]$	Upper triangular matrix
$\{s\}$	Associated displacement vector
$\{u\}$	Eigen Vector
$\{x\}$	Displacement Vector
$\{\dot{x}\}$	Velocity Vector
$\{\ddot{x}\}$	Acceleration Vector
$\{x\}^{(o)}$	Initial displacement vector
$\{\dot{x}\}^{(o)}$	Initial velocity vector
$\{\ddot{x}\}^{(o)}$	Initial acceleration vector
$\{y\}$	Associated displacement vector
A	Cross - sectional area of member
E	Young's modulus of elasticity
L	Length of member
P	Axial force
V	Shear force
W	Weight of the tributary wall areas

f	natural frequency of vibration
t	time
Δt	time interval
τ	natural period

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The introduction of the high speed electronic computer to field of numerical computation has revolutionized the approach to the analytical solution of many complicated problems. It has become particularly valuable to the field of engineering where specially prepared computer programs have been developed to aid in the solution of problems in structural analysis, stress analysis, surveying, fluid mechanics, machine design, vibrations and structural dynamics.

In most of the latter computer programs, the fundamental mathematical operations present are those related to matrix operations, including both matrix algebra and matrix calculus. From an engineering standpoint, one of the first basic text books related to the matrix methods is that written by Pipes[(1)*]. This text book covers the basic forms of matrix operation with applications to elasticity, dynamics, vibrations, and structural analysis. A second book by Pipes[(2)] published in 1969 presents a series of actual computer programs which may be utilized to efficiently solve many of the usual problems associated with

* number in parenthesis refers to literature cited in the bibliography

important matrix operations.

The purpose of this thesis is to develop a series of programs for suitable use on IBM 360-70 which is available at Youngstown State University which contain the matrix operation applicable to the solution of a typical engineering problem.

For each program formulated, a flow chart, a complete computer program in FORTRAN IV, and a sample of example illustrating the problem is presented for clarity and ease of interpretation.

The following list of computer programs are formulated:

1) Determinant Evaluation of a Matrix
2) Inversion Evaluation of a Matrix
3) Characteristic Equation Evaluation of a Matrix
4) Characteristic Value and Characteristic Vector Evaluation of a Matrix
5) Cholesky Transformation Evaluation for a Matrix.

The first four programs are standard problems in matrix operations which are essential to all matrix analysis procedures. The last program is a more recently developed technique in which a matrix is replaced by the product of an upper triangular and a lower triangular matrix. This technique is summarized by Westlake[3] and is specialized for the case of a symmetric matrix. Application of this technique to real engineering problems has been summarized by Parsons[4] in a master thesis at Youngstown State University.

The second section of this thesis presents a computer program solution for the analysis of a typical problem in Structural Dynamics. This solution consists of determining the dynamic response of a single-bay, multi-story, planar frame subjected to time varying forces. The method combines the use of finite difference techniques as reviewed by Rogers[5] simultaneously with matrix operations. The problem includes the modeling of the structural frame into a lumped-mass and spring mechanical system which generates a set of linear, coupled, total differential equation. The solution of these equations using classical scalar manual techniques is summarized by Fertis[6].

The computer solution developed in this work offers an efficient and economical means of determining the response of the structure for a variety of dynamic loading conditions and at the same time minimizes the time required to obtain these solutions.

CHAPTER II

GENERAL PROGRAMS

2.1 FORTRAN Program for Determinant Evaluation

This Determinant Evaluation program is based on the method of Chio(2)*. The idea of this method is to reduce the order of the determinant from higher order to lower order until a (1 x 1) determinant is obtained which gives the actual value of the original determinant. This method is illustrated using the following (4 x 4) determinant.

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} \qquad (2.1.1)$$

To reduce the order of the determinant from a (4 x 4) to a (3 x 3), any element of the determinant, say element (1,1), is made equal to unity by dividing the

* number in parenthesis refers to literature cited in the bibliography

first row through by a_{11}, yielding

$$D = a_{11} \begin{vmatrix} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \frac{a_{14}}{a_{11}} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{vmatrix} \tag{2.1.2}$$

Setting

$$a'_{12} = a_{12}/a_{11}$$
$$a'_{13} = a_{13}/a_{11}$$
$$a'_{14} = a_{14}/a_{11}$$

multiplying the first row by a_{21} and subtracting from the second row, then multiplying the first row by a_{31} and subtracting from the third row and, finally multiplying the first row by a_{41} and subtracting from the fourth row, gives

$$D' = a_{11} \begin{vmatrix} 1 & a'_{12} & a'_{13} & a'_{14} \\ 0 & a_{22} - a_{21}a'_{12} & a_{23} - a_{21}a'_{13} & a_{24} - a_{21}a'_{14} \\ 0 & a_{32} - a_{31}a'_{12} & a_{33} - a_{31}a'_{13} & a_{34} - a_{31}a'_{14} \\ 0 & a_{42} - a_{41}a'_{12} & a_{43} - a_{41}a'_{13} & a_{44} - a_{44}a'_{14} \end{vmatrix}$$

(2.1.3)

The value of this determinant now becomes

$$D'' = a_{11} (-1)^{1+1} \begin{vmatrix} a''_{11} & a''_{12} & a''_{13} \\ a''_{21} & a''_{22} & a''_{23} \\ a''_{31} & a''_{32} & a''_{33} \end{vmatrix} \qquad (2.1.4)$$

where $(-1)^{1+1}$ is the sign of element (1,1), and

$a''_{11} = a_{22} - a_{21} a'_{12}$ $\quad a''_{12} = a_{23} - a_{21} a'_{13}$ $\quad a''_{13} = a_{24} - a_{21} a'_{14}$

$a''_{21} = a_{32} - a_{31} a'_{12}$ $\quad a''_{22} = a_{33} - a_{31} a'_{13}$ $\quad a''_{23} = a_{34} - a_{31} a'_{14}$

$a''_{31} = a_{42} - a_{41} a'_{12}$ $\quad a''_{32} = a_{43} - a_{41} a'_{13}$ $\quad a''_{33} = a_{44} - a_{41} a'_{14}$

Thus, Equation (2.1.4) is a determinant of order (3 x 3). By repeating this operation, the determinant is reduced to a (2 x 2) and finally a (1 x 1) with the multipliers $a_{11}\, a''_{11}\, a'''_{11}$ ... , from which the value of the original (4 x 4) determinant is obtained.

The basic flow chart of this program is shown in Figure 1.

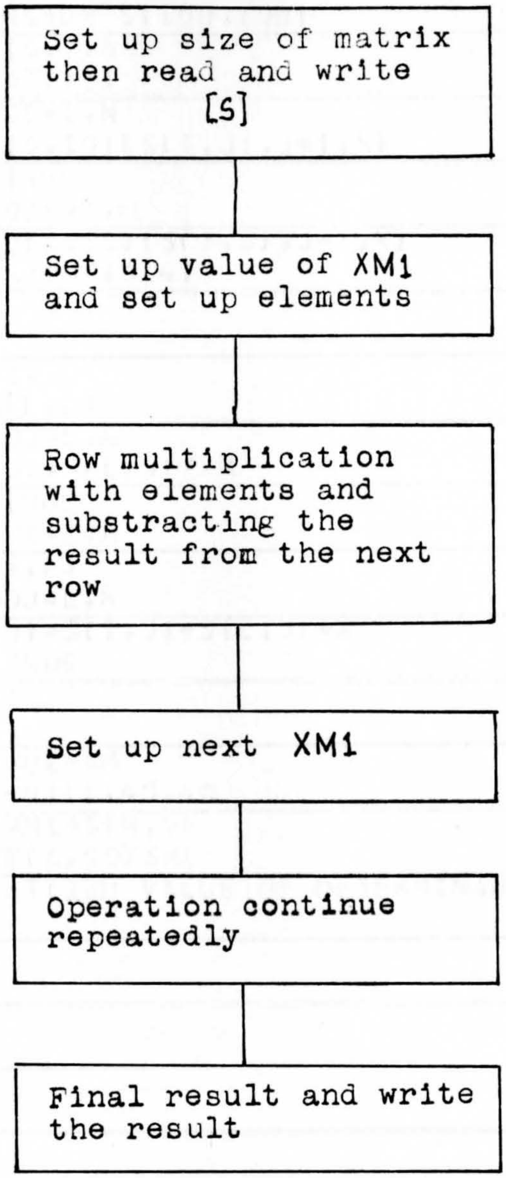

Figure 1 Flow chart of Determinant program

```
C        NAME OF THE PROGRAM=DETER
C        TO DETERMINE THE DETERMINANT OF THE MATRIX N*N
         DIMENSION S(100,100)
         READ(5,2)N
    2    FORMAT(I1)
         DO 90I=1,N
         READ(5,10)(S(I,J),J=1,N)
   90    CONTINUE
         DO 100I=1,N
  100    WRITE(6,10)(S(I,J),J=1,N)
   10    FORMAT(8F10.4)
         K=2
         L=1
         XM1=1.
   11    XM=S(L,L)
         DO 20J=L,N
         S(L,J)=S(L,J)/XM
   20    CONTINUE
         DO 30I=K,N
         X=S(I,L)
         DO 30J=L,N
         S(I,J)=S(I,J)-S(L,J)*X
   30    CONTINUE
         L=L+1
         K=K+1
         XM1=XM1*XM
         IF(L-N)11,40,40
   40    XM1=XM1*S(N,N)
         WRITE(6,50)XM1
   50    FORMAT(22H VALUE OF DETERMINANT= E10.3)
         STOP
         END
```

OPTIONS USED - PRINT,MAP,LET,CALL,NORES,NOTERM,SIZE=102400,NAME=**GO

NAME	TYPE	ADDR	NAME	TYPE	ADDR	NAME	TYPE	ADDR
MAIN	SD	150810	IHNECOMH*	SD	15A8B8	IBCOM# *	LR	15A8E4
IHNCOMH2*	SD	15B848	SEQDASD *	LR	15BC76	IHNFCVTH*	SD	15BF60
FCVLOUTP*	LR	15C09A	FCVZOUTP*	LR	15C1F6	FCVIOUTP*	LR	15C59E
INT6SWCH*	LR	15CA5C	IHNEFIOS*	SD	15CAD0	FIOCS# *	LR	15CAD0
IHNEFNTH*	SD	15E080	ARITH# *	LR	15EC80	ADJSWTCH*	LR	15E41C
ERRMON *	LR	15E930	IHNERRE *	LR	15E948	IHNFCONC*	SD	15EF30
FQCONI# *	LR	15F3D8	IHNUATBL*	SD	15F6C0	IHNETRCH*	SD	15F948
IHNFTEN *	SD	15FBF0	FTEN# *	LR	15FBF0			

TOTAL LENGTH F578
ENTRY ADDRESS 150810

```
  15.0000     1.0000    2.0000   -3.0000
   5.0000     6.0000    4.0000    4.0000
 -10.0000    -3.0000    2.0000    1.0000
  -5.0000     3.0000    4.0000    0.0
VALUE OF DETERMINANT=-0.182E+04
```

```
OPTIONS USED - PRINT,MAP,LET,CALL,NORES,NOTERM,SIZE=102400,NAME=**GO

   NAME   TYPE  ADDR           NAME    TYPE  ADDR           NAME    TYPE  ADDR

 MAIN       SD 180810      IHNECOMH*  SD 18A888      IBCOM#   *  LR 18A8E4
 IHNCOMH2*  SD 188848      SEQDASD *  LR 18BC76      IHNECVTH*   SD 18BE60
 FCVLOUTP*  LR 18C09A      FCVZOUTP*  LR 18C1F6      FCVIOUTP*   LR 18C59E
 INT6SWCH*  LR 18CA50      IHNEFIOS*  SD 18CAD0      FIOCS#   *  LR 18CAD0
 IHNEFNTH*  SD 18E080      ARITH#  *  LR 18E080      ADJSWTCH*   LR 18F41C
 ERRMON  *  LR 18E930      IHNERRE *  LR 18E948      IHNFCONO*   SD 18EF30
 FQCONI# *  LR 18F3D8      IHNUATBL*  SD 18F6C0      IHNETRCH*   SD 18F948
 IHNETEN *  SD 18F8E0      ETEN#   *  LR 18F8E0

 TOTAL LENGTH      F578
 ENTRY ADDRESS   180810

  1.0000    2.0000    3.0000
  3.0000    4.0000    6.0000
  2.0000    1.0000    1.0000
VALUE OF DETERMINANT=  0.100E+01
```

2.2 Program for the Inverse Matrix Evaluation

This matrix inversion program is formulated using the augmented matrix technique which is based on the Gauss-Jordan[2] method of solving simultaneous equations. This method involves the use of a unit matrix of the same order as the original matrix attached to the right hand side of the original matrix producing an (n x 2n) matrix. This new matrix is the augmented matrix in the form of Equation (2.2.1). Then, performing the proper matrix row operations, the original matrix is reduced to a unit matrix. The same row operations when applied to the attached unit matrix transform it into the inverse matrix

$$\left[\begin{array}{cccccc|cccccc} a_{11} & a_{12} & a_{13} & \cdot & \cdot & a_{1n} & 1 & 0 & 0 & \cdot & \cdot & 0 \\ a_{21} & a_{22} & a_{23} & \cdot & \cdot & a_{2n} & 0 & 1 & 0 & \cdot & \cdot & 0 \\ a_{31} & a_{32} & a_{33} & \cdot & \cdot & a_{3n} & 0 & 0 & 1 & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot & \cdot & & & \cdot \\ \cdot & \cdot & \cdot & & & \cdot & \cdot & \cdot & \cdot & & & \cdot \\ a_{n1} & a_{n2} & a_{n3} & \cdot & \cdot & a_{nn} & 0 & 0 & 0 & \cdot & \cdot & 1 \end{array}\right] \quad (2.2.1)$$

The general procedure includes dividing the first row by the leading coefficient, multiplying the first row by the leading coefficient of the second row, and then subtracting it from the second row. This procedure repeated for the third row through the n^{th} row yields

$$\left[\begin{array}{ccccc|ccccc} 1 & 0 & 0 & \cdots & 0 & b_{11} & b_{12} & b_{13} & \cdots & b_{1n} \\ 0 & 1 & 0 & \cdots & 0 & b_{21} & b_{22} & b_{23} & \cdots & b_{2n} \\ 0 & 0 & 1 & \cdots & 0 & b_{31} & b_{32} & b_{33} & \cdots & b_{3n} \\ \vdots & \vdots & \vdots & & & & & & & \\ 0 & 0 & 0 & \cdots & 1 & b_{n1} & b_{n2} & b_{n3} & \cdots & b_{nn} \end{array}\right] \tag{2.2.2}$$

The square matrix [B] in the right hand side of the Equation (2.2.2) is the inverse of the matrix [A]. The basic flow chart for the inversion matrix program is shown in Figure 2.

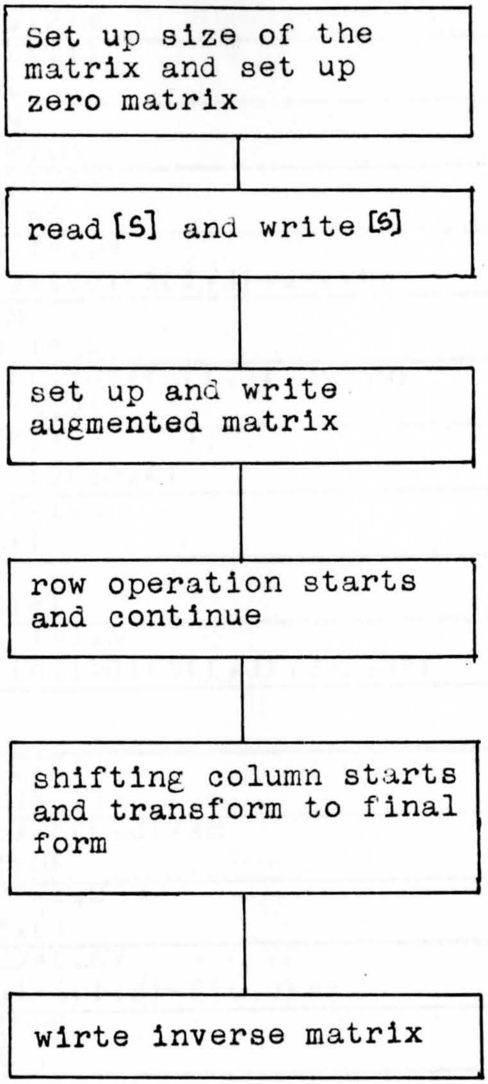

Figure 2 Flow chart of Inversion program

```
C        MATRIX INVERSION BY AUGMENTED MATRIX METHOD
         DIMENSION S(50,50)
         READ(5,2)N
  2      FORMAT(I1)
         NX=N+1
         NY=2*N
         DO  5I=1,N
         DO  5J=1,NY
  5      S(I,J)=0
         DO  10 I=1,N
         READ(5,100)(S(I,J),J=1,N)
  10     CONTINUE
         DO  20 I=1,N
  20     WRITE(6,100)(S(I,J),J=1,N)
         WRITE(6,21)
  21     FORMAT('          ')
  100    FORMAT(2F10.4)
         DO  30I=1,N
         NXX=N+I
         J=NXX
  30     S(I,J)=1
         DO  11 I=1,N
  11     WRITE(6,100)(S(I,J),J=1,NY)
         L=1
         K=2
  31     XM=S(L,L)
         DO  40J=L,NY
         S(L,J)=S(L,J)/XM
  40     CONTINUE
         DO  50I=K,N
         X=S(I,L)
         DO  50J=L,NY
         S(I,J)=S(I,J)-S(L,J)*X
  50     CONTINUE
         L=L+1
         K=K+1
         IF(L-N)31,31,51
  51     L=N
  52     LZ=L-1
         DO  60K=1,LZ
         I=L-K
         Y=S(I,L)
         DO  60J=L,NY
         S(I,J)=S(I,J)-S(L,J)*Y
  60     CONTINUE
         L=L-1
         IF(L-1)61,61,52
  61     WRITE(6,200)((S(I,J),J=NX,NY),I=1,N)
         WRITE(6,21)
  200    FORMAT(15H INVERSE MATRIX/(3X1P4E20.3))
         STOP
         END
```

OPTIONS USED - PRINT,MAP,LET,CALL,NORES,NOTERM,SIZE=102400,NAME=**GO

NAME	TYPE	ADDR	NAME	TYPE	ADDR	NAME	TYPE	ADDR
MAIN	SD	150810	IHNECOMH*	SD	15AB90	IBCOM# *	LR	15ABBC
IHNCOMH2*	SD	15BB20	SEQDASD *	LR	15BF4E	IHNFCVTH*	SD	15C238
FCVLOUTP*	LR	15C372	FCVZOUTP*	LR	15C4CE	FCVIOUTP*	LR	15C876
INT6SWCH*	LR	15CD28	IHNEFIOS*	SD	15CDA8	FIOCS# *	LR	15CDA8
IHNEFNTH*	SD	15E358	ARITH# *	LR	15E358	ADJSWTCH*	LR	15E6F4
ERRMON *	LR	15EC08	IHNERRE *	LR	15EC20	IHNFCONO*	SD	15F208
FQCONI# *	LR	15F6B0	IHNUATBL*	SD	15F998	IHNETRCH*	SD	15FC20
IHNFTEN *	SD	15FEC8	FTEN# *	LR	15FEC8			

TOTAL LENGTH F850
ENTRY ADDRESS 150810

```
  2.0000     1.0000     1.0000
  1.0000     3.0000     1.0000
  1.0000     1.0000     4.0000

  2.0000     1.0000     1.0000     1.0000     0.0        0.0
  1.0000     3.0000     1.0000     0.0        1.0000     0.0
  1.0000     1.0000     4.0000     0.0        0.0        1.0000
INVERSE MATRIX
          6.471E-01             -1.765E-01             -1.176E-01
         -1.765E-01              4.118E-01             -5.882E-02
         -1.176E-01             -5.882E-02              2.941E-01
```

OPTIONS USED - PRINT,MAP,LET,CALL,NORES,NOTERM,SIZE=102400,NAME=**GO

NAME	TYPE	ADDR	NAME	TYPE	ADDR	NAME	TYPE	ADD
MAIN	SD	150810	IHNECOMH*	SD	15AB90	IBCOM# *	LR	15AB
IHNCOMH2*	SD	15BB20	SEQDASD *	LR	15BF4E	IHNFCVTH*	SD	15C2
FCVLOUTP*	LR	15C372	FCVZOUTP*	LR	15C4CE	FCVIOUTP*	LR	15C8
INT6SWCH*	LR	15CD28	IHNEFIOS*	SD	15CDA8	FIOCS# *	LR	15CD
IHNEFNTH*	SD	15E358	ARITH# *	LR	15E358	ADJSWTCH*	LR	15E6
ERRMON *	LR	15EC08	IHNERRE *	LR	15EC20	IHNFCONO*	SD	15F2
FQCONI# *	LR	15F6B0	IHNUATBL*	SD	15F998	IHNETRCH*	SD	15FC
IHNFTEN *	SD	15FEC8	FTEN# *	LR	15FEC8			

TOTAL LENGTH F850
ENTRY ADDRESS 150810

15.0000	1.0000	2.0000	-3.0000
5.0000	6.0000	4.0000	4.0000
-10.0000	-3.0000	2.0000	1.0000
-5.0000	3.0000	4.0000	0.0

15.0000	1.0000	2.0000	-3.0000	1.0000	0.0	0.0
5.0000	6.0000	4.0000	4.0000	0.0	1.0000	0.0
-10.0000	-3.0000	2.0000	1.0000	0.0	0.0	1.0000
-5.0000	3.0000	4.0000	0.0	0.0	0.0	0.0

INVERSE MATRIX

4.615E-02	3.077E-02	1.538E-02
-8.791E-02	-1.099E-02	-2.198E-01
1.236E-01	4.670E-02	1.841E-01
-4.945E-02	1.813E-01	1.264E-01

2.3 FORTRAN Program for the Characteristic Equation Problem

The method of Danilevsky[2] for obtaining the characteristic equation is based on the reduction of matrix ,

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ \cdot & \cdot & \cdot & & \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn} \end{bmatrix} \qquad (2.3.1)$$

whose characteristic equation is desired, to matrix [P] of the form

$$[P] = \begin{bmatrix} p_1 & p_2 & p_3 & \cdots & p_{n-1} & p_n \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & & & \\ \cdot & \cdot & \cdot & & & \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \qquad (2.3.2)$$

The elements $p_1, p_2, p_3 \ldots, p_n$ represent the coefficients of the characteristic equation of matrix[P] . The process is accomplished by a series of similarity transformations. Since the characteristic equations of similar matrices are identical, the characteristic equations of matrices [A] and [P] are the same. Coefficients of the characteristic polynomial of matrix [P] is given by the determinant

$$D(\lambda) = \left| [P] - \lambda [I] \right| = \left| \begin{bmatrix} p_1 - \lambda & p_2 & p_3 & \cdots & p_n \\ 1 & -\lambda & 0 & \cdots & 0 \\ 0 & 1 & -\lambda & \cdots & 0 \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ 0 & 0 & 0 & & -\lambda \end{bmatrix} \right| \qquad (2.3.3)$$

which expanded in terms of the elements of the first row, gives

$$D(\lambda) = (-1)^n (\lambda^n - p_1 \lambda^{n-1} - p_2 \lambda^{n-2} \ldots - p_n) = 0$$

where $D(\lambda)$ is the desired characteristic polynomial of both matrix [A] and [P] and n is the order of the square matrix [A] .

The transformation from matrix [A] to matrix [P] is performed on the following (4 x 4) matrix

$$[A] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \qquad (2.3.4)$$

Matrix [M] is first formed from the elements of the fourth row of matrix [A] as

$$[M] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{a_{41}}{a_{43}} & -\frac{a_{42}}{a_{43}} & -\frac{1}{a_{43}} & \frac{a_{44}}{a_{43}} \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (2.3.5)$$

The product of [A] and [M] become

$$[A][M] = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{a_{41}}{a_{43}} & -\frac{a_{42}}{a_{43}} & \frac{1}{a_{43}} & -\frac{a_{44}}{a_{43}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

or

$$[A][M] = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \qquad (2.3.6)$$

where

$b_{11} = a_{11} - \frac{a_{13}a_{14}}{a_{43}}$ $\qquad b_{12} = a_{12} - \frac{a_{13}a_{42}}{a_{43}}$ $\qquad b_{41} = 0$

$b_{13} = a_{23}/a_{43}$ $\qquad b_{14} = -a_{13}a_{44}/a_{43}$ $\qquad b_{42} = 0$

$b_{21} = a_{21} - \frac{a_{23}a_{41}}{a_{43}}$ $\qquad b_{22} = a_{22} - \frac{a_{23}a_{42}}{a_{43}}$ $\qquad b_{43} = 1$

$b_{23} = a_{23}/a_{43}$ $\qquad b_{24} = a_{24} - \frac{a_{23}a_{44}}{a_{43}}$ $\qquad b_{44} = 0$

$b_{31} = a_{31} - \frac{a_{33}a_{41}}{a_{43}}$ $\qquad b_{32} = a_{32} - \frac{a_{33}a_{42}}{a_{43}}$

$b_{33} = a_{33}/a_{34}$ $\qquad b_{34} = a_{34} - \frac{a_{33}a_{44}}{a_{43}}$

The matrix of Equation (2.3.6) is not yet similar to matrix [A]. It is made similar to [A] by premultiplying Equation (2.3.6) by the inverse of matrix [M] which is obtained from Equation (2.3.5) as

$$[M]^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \\ 0 & 0 & 0 & 0 \end{bmatrix} \qquad (2.3.7)$$

Thus, the similarity transformation of [A] takes the form

$$[C] = [M]^{-1}[A][M] \qquad (2.3.8\ a)$$

$$[C] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ -\frac{a_{41}}{a_{43}} & -\frac{a_{42}}{a_{43}} & \frac{1}{a_{43}} & -\frac{a_{44}}{a_{43}} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[C] = \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (2.3.8\ b)$$

The process is continued in reducing the matrix of Equation (2.4.8b) to the form

$$[D] = \begin{bmatrix} 1 & 0 & 0 & 0 \\ c_{31} & c_{32} & c_{33} & c_{34} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_{11} & c_{12} & c_{13} & c_{14} \\ c_{21} & c_{22} & c_{23} & c_{24} \\ c_{31} & c_{32} & c_{33} & c_{34} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ -\frac{c_{31}}{c_{32}} & \frac{1}{c_{32}} & -\frac{c_{33}}{c_{32}} & -\frac{c_{34}}{c_{32}} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[D] = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} \\ d_{21} & d_{22} & d_{23} & d_{24} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (2.3.9)$$

The process is repeated a step further which reduces the (4 x 4) matrix of Equation (2.3.4) to the form of Equation (2.3.2) which simplifies to

$$[P] = \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \qquad (2.3.10)$$

where the p's represent the coefficients of the characteristic polynomial of Equation (2.3.4). Then, the characteristic polynomial equation is written as

$$D(\lambda) = (-1)^n(\lambda^n - p_1\lambda^{n-1} - p_2\lambda^{n-2} - p_3\lambda^{n-3} - p_4) = 0$$

The flow chart of the computer program for the characteristic equation is shown in Figure 3.

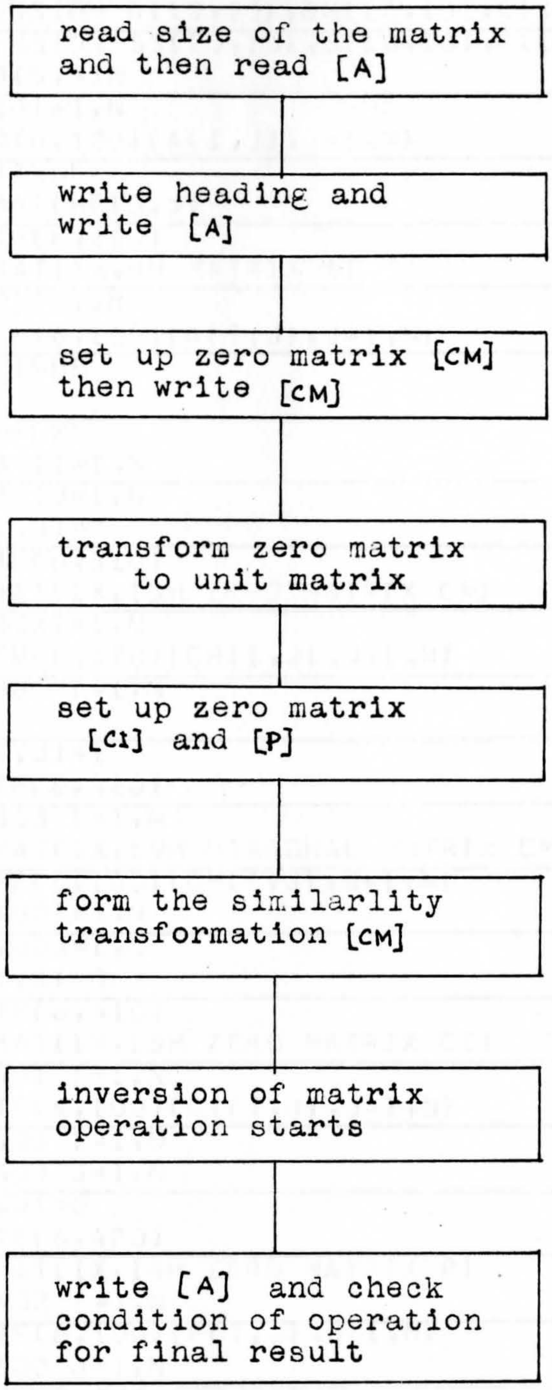

Figure 3 Flow chart of Characteristic Equation

```
C     THIS PROGRAM USE TO COMPUTE THE CHARACTERISTIC EQUATION
C     BY USING A. M. DANILEVSKY METHOD
      DIMENSION A(10,10),CM(10,10),CI(10,10),P(10,10)
      DIMENSION SS(10,10),S(10,10),R(10,10)
      READ(5,*)N
      DO 10I=1,N
      READ(6,100)(A(I,J),J=1,N)
10    CONTINUE
100   FORMAT(8F10.5)
      WRITE(6,110)
110   FORMAT(1X,9H MATRIX A)
      DO 20I=1,N
      WRITE(6,100)(A(I,J),J=1,N)
20    CONTINUE
      LX=1
      LX=N-LX
997   DO 301I=1,N
      DO 301J=1,N
301   CM(I,J)=0
      WRITE(6,310)
310   FORMAT(1X,15H ZERO MATRIX CM)
      DO 302I=1,N
302   WRITE(6,100)(CM(I,J),J=1,N)
      DO 300 I=1,N
      J=I
300   CM(I,J)=1
      WRITE(6,320)
      DO 303 I=1,N
320   FORMAT(1X,19H DIAGONAL MATRIX CM,/)
303   WRITE(6,100)(CM(I,J),J=1,N)
      DO 400I=1,N
      DO 400J=1,N
400   CI(I,J)=0
      WRITE(6,410)
410   FORMAT(1X,15H ZERO MATRIX CI)
      DO 401 I=1,N
401   WRITE(6,100)(CI(I,J),J=1,N)
      DO 403 I=1,N
      DO 403 J=1,N
403   P(I,J)=0
      WRITE(6,420)
420   FORMAT(1X,14H ZERO MATRIX P)
      DO 402 I=1,N
402   WRITE(6,100)(P(I,J),J=1,N)
      DO 500 J=1,N
      IF(J-LX)520,530,520
520   CM(LX,J)=-A(LX+1,J)/A(LX+1,LX)
      GO TO 500
530   CM(LX,J)=1/A(LX+1,LX)
500   CONTINUE
      WRITE(6,510)
510   FORMAT(1X,14H NEW MATRIX CM)
      DO 600I=1,N
600   WRITE(6,100)(CM(I,J),J=1,N)
      DO 700 I=1,N
```

```
      DO 710 J=1,N
      S(I,J)=CM(I,J)
  710 CONTINUE
  700 CONTINUE
      WRITE(6,540)
  540 FORMAT(1X,13H NEW MATRIX S)
      DO 720 I=1,N
  720 WRITE(6,100)(S(I,J),J=1,N)
      NX=N+1
      NY=2*N
      DO 5 I=1,N
      DO 5 J =1,N
    5 S(I,J)=0
      DO 32I=1,N
      DO 32J=NX,NY
   32 S(I,J)=0
      DO 6 I=1,N
      DO 7 J=1,N
      S(I,J)=CM(I,J)
    7 CONTINUE
    6 CONTINUE
      WRITE(6,540)
      DO 30I=1,N
      NXX=N+I
      J=NXX
   30 S(I,J)=1
      DO 11 I=1,N
   11 WRITE(6,100)(S(I,J),J=1,NY)
      L=1
      K=2
   31 XM=S(L,L)
      DO 40J=L,NY
      S(L,J)=S(L,J)/XM
   40 CONTINUE
      DO 50I=K,N
      X=S(I,L)
      DO 50J=L,NY
      S(I,J)=S(I,J)-S(L,J)*X
   50 CONTINUE
      L=L+1
      K=K+1
      IF(L-N)31,31,51
   51 L=N
   52 LZ=L-1
      DO 60K=1,LZ
      I=L-K
      Y=S(I,L)
      DO 60J=L,NY
      S(I,J)=S(I,J)-S(L,J)*Y
   60 CONTINUE
      L=L-1
      IF(L-1)61,61,52
   61 WRITE(6,200)((S(I,J),J=NX,NY),I=1,N)
  200 FORMAT(15H INVERSE MATRIX/(3X1P4E20.6))
      DO 555 I=1,N
```

```
      DO 555J=1,N
555   P(I,J)=0
      DO 666 M=1,N
      DO 70I=1,N
      DO 80K=1,N
80    P(M,I)=A(M,K)*CM(K,I)+P(M,I)
70    CONTINUE
666   CONTINUE
      WRITE(6,222)
222   FORMAT(15H PRODUCT MATRIX)
      DO 90I=1,N
      WRITE(6,101)(P(I,J),J=1,N)
90    CONTINUE
      DO 55 I=1,N
      DO 55 J=1,N
55    A(I,J)=0
      DO 56 M=1,N
      DO 57 I=1,N
      J=0
      DO 58 K=NX,NY
      J=J+1
58    A(M,I)=S(M,K)*P(J,I)+A(M,I)
57    CONTINUE
56    CONTINUE
      WRITE(6,222)
      DO 59I=1,N
59    WRITE(6,101)(A(I,J),J=1,N)
101   FORMAT(3X1P4E20.6)
      LX=LX-1
      IF(LX)999,999,998
998   GO TO 997
999   STOP
      END
```

MATRIX A

-5.50988	1.87009	0.42291	0.00881
0.28786	-11.81170	5.71190	0.05872
0.04910	4.30803	-12.07070	0.22933
0.00623	0.26985	1.39737	-17.59621

INVERSE MATRIX

1.000000E+00	0.0	0.0	0.0
0.0	1.000000E+00	0.0	0.0
6.234996E-03	2.698510E-01	1.397370E+00	-1.759619E-02
0.0	0.0	0.0	1.000000E+00

PRODUCT MATRIX

-5.511766E+00	1.788420E+00	3.026456E-01	5.334228E+04
2.623788E-01	-1.291474E+01	4.087605E+00	7.198506E-04
1.859188E-01	6.046163E+00	-2.946194E+01	-2.084559E-05
3.725290E-09	5.960464E-08	9.999999E-01	-1.525879E-05

INVERSE MATRIX

1.000000E+00	0.0	0.0	0.0
1.859188E-01	6.046164E+00	-2.946194E+01	-2.082558E+02
0.0	0.0	1.000000E+00	0.0
0.0	0.0	0.0	1.000000E+00

PRODUCT MATRIX

-5.566760E+00	2.957941E-01	9.017315E+00	6.699422E+01
2.952511E+00	-4.232167E+01	-5.625576E+02	-2.244458E+03
5.960464E-08	9.999998E-01	-1.525879E-05	-9.155273E-05
1.892454E-09	9.858258E-09	1.000000E+00	-1.320378E-05

INVERSE MATRIX

2.952511E+00	-4.232167E+01	-5.625576E+02	-2.244458E+03
0.0	1.000000E00	0.0	0.0
0.0	0.0	1.000000E+00	0.0
0.0	0.0	0.0	1.000000E+00

PRODUCT MATRIX

-4.788841E+01	-7.972781E+02	-5.349441E+03	-1.229648E+04
9.999999E-01	-1.625879E-05	-2.441406E-04	-2.441406E-04
2.018788E-08	1.000000E+00	-3.901999E-06	-4.624210E-05
6.409642E-10	3.698494E-08	1.000000E+00	-1.176516E-05

2.4 Program for Evaluation of Characteristic Values and Characteristic Vectors

The characteristic-value, characteristic-vector problem is an extremely important one since the dynamic behavior of a linear mechanical systems are directly predictable by its usage.

Consider the vector matrix equation

$$\{y\} = [A]\{x\} \tag{2.4.1}$$

where $\{y\}$ and $\{x\}$ are column vectors and $[A]$ is a square matrix (n x n). This equation may be viewed as a transformation of the vector $\{x\}$ into the vector $\{y\}$. The dilitation transformation maps the vector $\{x\}$ into a constant times itself; it follows that

$$\{y\} = \lambda [I] \{x\} \tag{2.4.2}$$

or

$$\Big[[A] - \lambda [I]\Big] \{x\} = \{0\} \tag{2.4.3}$$

where λ is defined as the characteristic value and $\{x\}$ is defined as the associated characteristic vector. The homogeneous Equation (2.4.3) has a solution which exists if and only if, the following determinant equation holds:

$$\det\left[[A]-\lambda[I]\right] = 0 \tag{2.4.4}$$

for n = 3.

$$\begin{vmatrix} a_{11}-\lambda & a_{12} & a_{13} \\ a_{21} & a_{22}-\lambda & a_{23} \\ a_{31} & a_{32} & a_{33}-\lambda \end{vmatrix} = 0 \tag{2.4.5}$$

This equation of degree 3 (generally of degree n) for λ is called characteristic equation and takes the form

$$\lambda^3 - I_1\lambda^2 + I_2\lambda - I_3 = 0 \tag{2.4.6}$$

For each of the roots λ_i, Equation (2.4.3) has a solution $\{x\} \neq 0$ called the characteristic vector of $[A]$. The characteristic values and characteristic vectors solutions are programed by using an iteration process. The characteristic value and the characteristic vectors are related as follows:

$$\left[[A]-\lambda[I]\right]\{\Delta\} = \{0\} \tag{2.4.7}$$

This matrix equation can be written as

$$\begin{bmatrix} a_{11}-\lambda & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22}-\lambda & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33}-\lambda & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nn}-\lambda \end{bmatrix} \begin{Bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{Bmatrix}$$

(2.4.8)

where λ is the characteristic value, a_{ij}'s are the elements of matrix [A] , and s_i's are the elements of the characteristic vector corresponding to the value λ. The characteristic values and characteristic vectors can be obtained by method of iteration process. Using Equation (2.4.8) form

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ a_{1n} & a_{2n} & a_{3n} & & a_{nn} \end{bmatrix} \begin{Bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{Bmatrix} = \lambda \begin{Bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{Bmatrix}$$

(2.8.9)

In the matrix iteration method, an arbitrary set of s_i (i.e. s_1, s_2, s_3 ... s_n) are used to initiate the problem. For convenience in calculations, s_n is taken equal to unity. Equation (2.4.8) forms a set of homogeneous equations, hence, the absolute values of s's can not be determined. However, the ratios of s's may be obtained. The initial set of s_i, $i = 1,2,3, \ldots n$, are substituted into the left hand side of Equation (2.4.9) performing the indicated matrix multiplication on the left hand side of Equation (2.4.9), the vector on the right hand side is calculated. This vector is factored by defining the new value of s_n as $s_n = \lambda(1)$. This value λ is factored from each of the remaining vector components. The value λ is the first approximation to the characteristic value and the factored vector is the first approximation to the characteristic vector. The method proceeds by taking the next approximation for the vector solution as the previous solution. It is necessary to iterate a number of times in order to improve accuracy.

Continuing this process, the iteration converges, resulting in the characteristic value λ and the corresponding characteristic vector. The rate of convergence of this iteration process depends on the numerical seperation of the characteristic value of matrix [A]. It can also be shown that the characteristic value obtained by this method is the largest characteristic value or equivalently the largest root of the characteristic equation. For the special of a symmetric matrix [A], the characteristic values are always real, and the characteristic vectors are always orthogonal.

The basic flow chart for the program of obtaining characteristic value and characteristic vector by iteration process is shown in Figure 4.

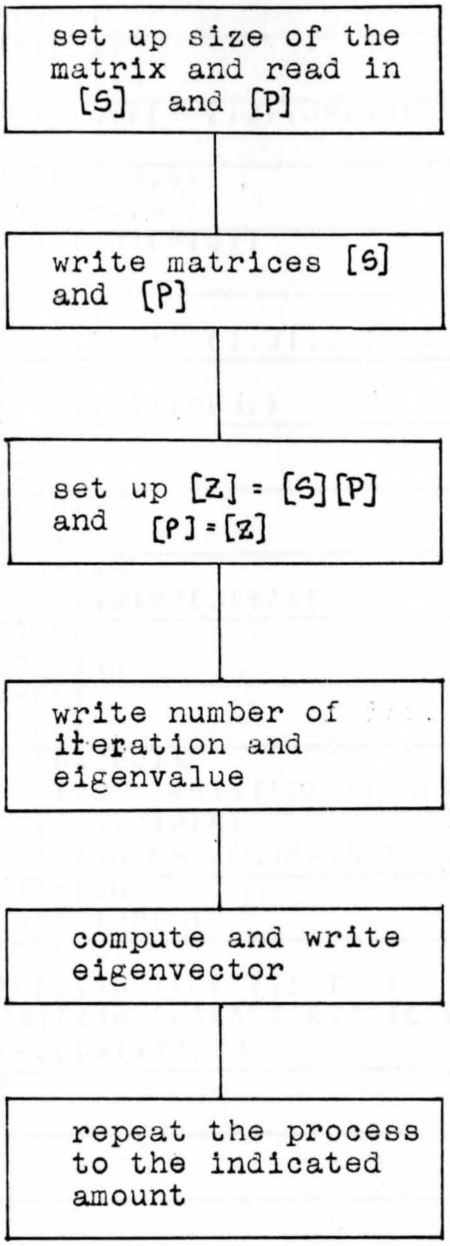

Figure 4 Flow chart of Characteristic value and Characteristic vector

```
C       PROGRAM TO EVALUATE EIGENVALUE & EIGENVECTOR
        DIMENSION S(100,100),P(100),Z(100)
        READ(5,2)N
2       FORMAT(I1)
        DO 10 I=1,N
        READ(5,100)(S(I,J),J=1,N)
10      CONTINUE
100     FORMAT(8F10.4)
        DO 30  J=1,N
        READ(5,100)(P(J))
30      CONTINUE
        DO 20 I=1,N
20      WRITE(6,100)(S(I,J),J=1,N)
        DO 40 J=1,N
40      WRITE(6,100)(P(J))
        K=0
41      K=K+1
        DO 50I=1,N
        Z(I)=0.
        DO 50J=1,N
        Z(I)=S(I,J)*P(J)+Z(I)
50      CONTINUE
        DO 60I=1,N
        P(I)=Z(I)
60      CONTINUE
        WRITE(6,200)K
200     FORMAT(20H IRETATION NUMBER=  I6)
        WRITE(6,300)P(N)
300     FORMAT(25H CHARACTERISTIC VALUE=     E10.5)
        DO 61I=1,N
        P(I)=P(I)/P(N)
61      CONTINUE
        WRITE(6,400)(P(I),I=1,3)
400     FORMAT(23H CHARACTERISTIC VECTOR /(6XE15.5))
        IF(K-20)41,70,70
70      STOP
        END
```

```
 2.        -1.        0.
-1.         2.       -1.
 0.        -1.        1.

 1.
-1.
 1.

  2.0000   -1.0000    0.0
 -1.0000    2.0000   -1.0000
  0.0      -1.0000    1.0000

  1.0000
 -1.0000
  1.0000
IRETATION NUMBER=          1
CHARACTERISTIC VALUE=        0.200E+01
CHARACTERISTIC VECTOR
       0.150E+01
      -0.200E+01
       0.100E+01
IRETATION NUMBER=          2
CHARACTERISTIC VALUE=        0.300E+01
CHARACTERISTIC VECTOR
       0.167E+01
      -0.217E+01
       0.100E+01
IRETATION NUMBER=          3
CHARACTERISTIC VALUE=        0.317E+01
CHARACTERISTIC VECTOR
       0.174E+01
      -0.221E+01
       0.100E+01
IRETATION NUMBER=          4
CHARACTERISTIC VALUE=        0.321E+01
CHARACTERISTIC VECTOR
       0.177E+01
      -0.223E+01
       0.100E+01
IRETATION NUMBER=          5
CHARACTERISTIC VALUE=        0.323E+01
CHARACTERISTIC VECTOR
       0.179E+01
      -0.224E+01
       0.100E+01
IRETATION NUMBER=          6
CHARACTERISTIC VALUE=        0.324E+01
CHARACTERISTIC VECTOR
       0.179E+01
      -0.224E+01
       0.100E+01
IRETATION NUMBER=          7
CHARACTERISTIC VALUE=        0.324E+01
CHARACTERISTIC VECTOR
       0.180E+01
      -0.225E+01
```

```
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=        9
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       10
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       11
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       12
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       13
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       14
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       15
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       16
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       17
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=       18
CHARACTERISTIC VALUE=      0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
```

```
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
IRETATION NUMBER=      20
CHARACTERISTIC VALUE=     0.325E+01
CHARACTERISTIC VECTOR
      0.180E+01
     -0.225E+01
      0.100E+01
R;
```

2.5 Program for the Method of Cholesky Transformation

The Cholesky process is particularly useful in problems of structural dynamics[4]. This scheme for the solution of a system of linear equations related to structural analysis is very desirable. Theoretically, Cholesky's method is based on the fact that any square matrix may be expressed as the product of an upper-triangular matrix and a lower-triangular matrix.

Let [A] be a non symmetric square matrix, [U] an upper triangular matrix, and [L] a lower triangular matrix, such that they are related by the following general matrix equation[3]:

$$[A] = [L][U] \qquad (2.5.1)$$

If the square matrix [A] is a (4 x 4) matrix, then

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ l_{21} & 1 & 0 & 0 \\ l_{31} & l_{32} & 1 & 0 \\ l_{41} & l_{42} & l_{43} & 1 \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

(2.5.2)

The expansion of the above equation yields a set of sixteen equations from which the values of the elements of matrices [L] and [U] are obtained as functions of the elements of the matrix [A] . To illustrate the procedure, consider the following partial set of equation:

$$a_{11} = u_{11}$$

$$a_{21} = l_{21} u_{11} \; ; \qquad l_{21} = \frac{a_{21}}{u_{11}} = \frac{a_{21}}{a_{11}}$$

$$a_{31} = l_{31} u_{11} \; ; \qquad l_{31} = \frac{a_{31}}{u_{11}} = \frac{a_{31}}{a_{11}}$$

$$a_{41} = l_{41} u_{11} \; ; \qquad l_{41} = \frac{a_{41}}{u_{11}} = \frac{a_{41}}{a_{11}}$$

$$a_{12} = u_{12}$$

$$a_{22} = l_{21} u_{12} + u_{22} \; ; \qquad u_{22} = a_{22} - l_{21} u_{12} = a_{22} - \frac{a_{21}}{a_{11}} \cdot a_{12}$$

$$a_{32} = l_{31} u_{12} + l_{32} u_{22} \; ; \qquad l_{32} = \frac{(a_{32} - l_{21} u_{12})}{u_{22}}$$

$$a_{42} = l_{41} u_{12} + l_{42} u_{22} \; ; \qquad l_{42} = \frac{(a_{42} - l_{41} u_{12})}{u_{22}}$$

The remaining eight values of the ℓ's and the u's are obtained from the additional remaining equations.

Cholesky's method offers additional advantages for matrices which are symmetric. A symmetric matrix [A] may be written as the product of two triangular matrices, one of them being the transpose of the other in the form

$$[A] = [U]^T[U] \qquad (2.5.3a)$$

which is expanded to

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{12} & a_{22} & a_{23} & a_{24} \\ a_{13} & a_{23} & a_{33} & a_{34} \\ a_{14} & a_{24} & a_{34} & a_{44} \end{bmatrix} = \begin{bmatrix} u_{11} & 0 & 0 & 0 \\ u_{12} & u_{22} & 0 & 0 \\ u_{13} & u_{23} & u_{33} & 0 \\ u_{14} & u_{24} & u_{34} & u_{44} \end{bmatrix} \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ 0 & u_{22} & u_{23} & u_{24} \\ 0 & 0 & u_{33} & u_{34} \\ 0 & 0 & 0 & u_{44} \end{bmatrix}$$

(2.5.3b)

For the (4 x 4) sysmetrix matrix shown, the ten elements of the matrix [U] are obtained by the multiplication of the matrices of Equation (2.5.3b) and in part become

$$a_{11} = u_{11}^2 ; \qquad u_{11} = (a_{11})^{1/2}$$

$$a_{12} = u_{11} u_{12} ; \qquad u_{12} = a_{12} \cdot \frac{1}{u_{11}} \quad = \frac{a_{12}}{(a_{11})^{1/2}}$$

$$a_{13} = u_{11} u_{13} ; \qquad u_{13} = a_{13} \cdot \frac{1}{u_{11}} \quad = \frac{a_{13}}{(a_{11})^{1/2}}$$

$$a_{22} = u_{12}^2 + u_{22}^2 ; \qquad u_{22} = (a_{22} - u_{12})^{1/2} = \left[a_{22} - \frac{a_{12}^2}{a_{11}}\right]^{1/2}$$

(2.5.3c)

These operations are generalized for an (n x n) matrix by the mathematical expression

$$u_{ij} = \frac{a_{ij} - \sum_{k=1}^{i-1} u_{ki} u_{kj}}{u_{ii}} \qquad \text{for } \begin{array}{l} j = i+1, \ldots, n \\ i = 2, \ldots, n \end{array} \tag{2.5.4}$$

$$u_{ij} = \frac{a_{ij}}{u_{ii}} \qquad \text{for } \begin{array}{l} i = 1 \\ j = 2, 3, \ldots, n \end{array} \tag{2.5.6}$$

$$u_{ii} = (a_{ii})^{1/2} \qquad \text{for } i = 1 \tag{2.5.7}$$

$$u_{ii} = (a_{ii} - \sum_{k=1}^{i-1} u_{ki}^2)^{1/2} \qquad \text{for } i = 2, \ldots, n \tag{2.5.8}$$

The flow chart of the computer program for Cholesky triangulation is shown in Figure 5.

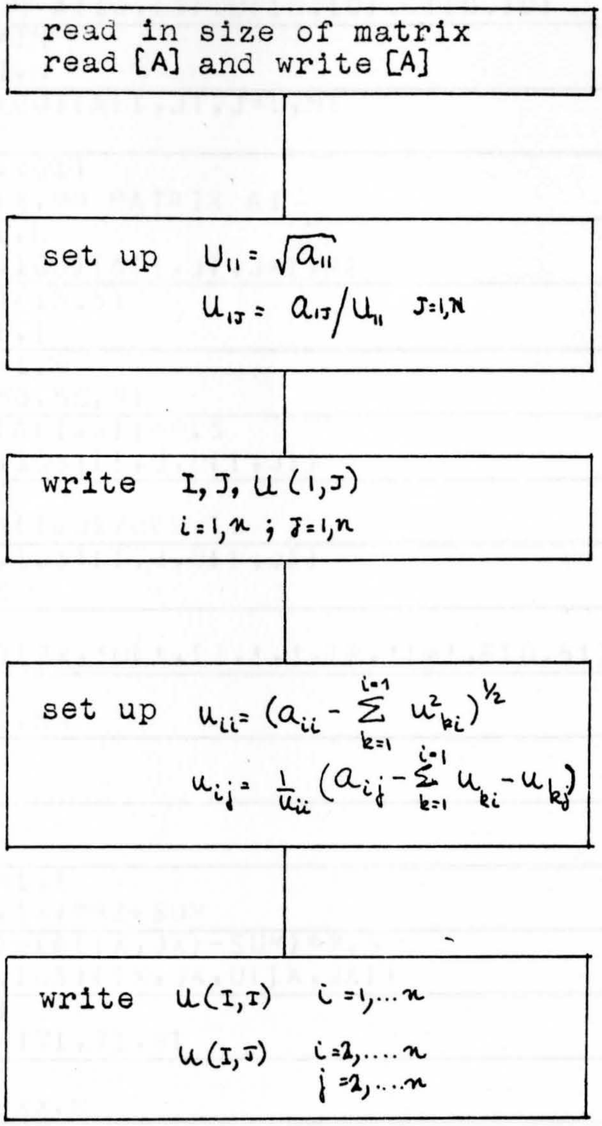

Figure 5 Flow chart of Cholesky Triangulation

```
C        THIS PROGRAM WILL COMPUTE U & U TRANSPOSE TRANSPOSED-MATRIX FROM
C        THE GIVEN MATRIX
         DIMENSION A(10,10),U(10,10),C(10,10)
         READ(5,*)N
         DO  10I=1,N
         READ(5,100)(A(I,J),J=1,N)
   10    CONTINUE
         WRITE(6,101)
  101    FORMAT(1X,9H MATRIX A)
         DO  20I=1,N
   20    WRITE(6,100)(A(I,J),J=1,N)
  100    FORMAT(8F10.5)
         DO  30I=1,1
         DO  40 J=1,N
         IF(J-1)50,50,51
   50    U(I,J)=(A(I,J))**.5
         WRITE(6,103)(I,J,U(I,J))
         GO TO 40
   51    U(I,J)=A(I,J)/U(I,I)
         WRITE(6,103)(I,J,U(I,J))
   40    CONTINUE
   30    CONTINUE
  103    FORMAT(1(2X,'U(',I2,',',I2,')=',F10.5))
         NN=N-1
         DO  60I=1,NN
         SUM=0
         J=I
         IX=I+1
         JX=J+1
         DO  70 K=1,I
   70    SUM=U(K,IX)**2+SUM
         U(IX,JX)=(A(IX,JX)-SUM)**.5
         WRITE(6,103)(IX,JX,U(IX,JX))
         JXX=JX+1
         IF(JXX-N)71,71,91
   71    CONTINUE
         DO  75M=JXX,N
         SUM=0
         DO  80L=1,I
   80    SUM=U(L,JX)*U(L,M)+SUM
         U(JX,M)=(A(IX,M)-SUM)/U(IX,JX)
   75    WRITE(6,103)(JX,M,U(JX,M))
   60    CONTINUE
   91    CONTINUE
         WRITE(6,104)
  104    FORMAT(1X,9H MATRIX U)
         DO  90I=1,N
   90    WRITE(6,100)(U(I,J),J=1,N)
         STOP
         END
```

OPTIONS USED - PRINT,MAP,LET,CALL,NORES,NOTERM,SIZE=102400,NAME=**G

NAME	TYPE	ADDR	NAME	TYPE	ADDR	NAME	TYPE	AD
MAIN	SD	150810	IHNECOMH*	SD	151360	IBCOM# *	LR	151
IHNCOMH2*	SD	1522F0	SEQDASD *	LR	15271E	IHNLDFIO*	SD	152
IHNFRXPR*	SD	153BC8	FRXPR# *	LR	153BC8	IHNFCVTH*	SD	153
FCVLOUTP*	LR	153E8A	FCVZOUTP*	LR	153FE6	FCVIOUTP*	LR	154
INT6SWCH*	LR	154840	IHNEFIOS*	SD	1548C0	FIOCS# *	LR	154
IHNEFNTH*	SD	155E70	ARITH# *	LR	155E70	ADJSWTCH*	LR	156
ERRMON *	LR	156720	IHNERRE *	LR	156738	IHNUATBL*	SD	156
ALOG *	LR	156FC0	IHNSEXP *	SD	157180	EXP *	LR	157
IHNFCONI*	SD	1577D8	FQCONI# *	LR	1577D8	IHNETRCH*	SD	157
IHNFTEN *	SD	157D68	FTEN# *	LR	157D68			

TOTAL LENGTH 76F0
ENTRY ADDRESS 150810

MATRIX A

1.00000	2.00000	3.00000	2.00000	1.00000
2.00000	5.00000	8.00000	7.00000	6.00000
3.00000	8.00000	17.00000	14.00000	15.00000
2.00000	7.00000	14.00000	23.00000	28.00000
1.00000	6.00000	15.00000	28.00000	62.00000

```
U( 1, 1)=   1.00000
U( 1, 2)=   2.00000
U( 1, 3)=   3.00000
U( 1, 4)=   2.00000
U( 1, 5)=   1.00000
U( 2, 2)=   1.00000
U( 2, 3)=   2.00000
U( 2, 4)=   3.00000
U( 2, 5)=   4.00000
U( 3, 3)=   2.00000
U( 3, 4)=   1.00000
U( 3, 5)=   2.00000
U( 4, 4)=   3.00000
U( 4, 5)=   4.00000
U( 5, 5)=   5.00000
```

MATRIX U

1.00000	2.00000	3.00000	2.00000	1.00000
0.0	1.00000	2.00000	3.00000	4.00000
0.0	0.0	2.00000	1.00000	2.00000
0.0	0.0	0.0	3.00000	4.00000
0.0	0.0	0.0	0.0	5.00000

OPTIONS USED - PRINT,MAP,LET,CALL,NORES,NOTERM,SIZE=102400,NAME=**GO

NAME	TYPE	ADDR	NAME	TYPE	ADDR	NAME	TYPE	ADDR
MAIN	SD	150810	IHNECOMH*	SD	151360	IBCOM# *	LR	15138C
IHNCOMH2*	SD	1522F0	SEQDASD *	LR	15271E	IHNLDFIO*	SD	152A08
IHNFRXPR*	SD	153BC8	FRXPR# *	LR	153BC8	IHNFCVTH*	SD	153D50
FCVLOUTP*	LR	153E8A	FCVZOUTP*	LR	153FE6	FCVIOUTP*	LR	15438E
INT6SWCH*	LR	154840	IHNEFIOS*	SD	1548C0	FIOCS# *	LR	1548C0
IHNEFNTH*	SD	155E70	ARITH# *	LR	155E70	ADJSWTCH*	LR	15620C
ERRMON *	LR	156720	IHNERRE *	LR	156738	IHNUATBL*	SD	156D20
ALOG *	LR	156FC0	IHNSEXP *	SD	157180	EXP *	LR	157180
IHNFCONI*	SD	1577D8	FQCONI# *	LR	1577D8	IHNETRCH*	SD	157ACC
IHNFTEN *	SD	157D68	FTEN# *	LR	157D68			

TOTAL LENGTH 76F0
ENTRY ADDRESS 150810

MATRIX A

1.00000	2.00000	3.00000	2.00000	3.00000
2.00000	8.00000	8.00000	8.00000	8.00000
3.00000	8.00000	14.00000	12.00000	12.00000
2.00000	8.00000	12.00000	21.00000	16.00000
3.00000	8.00000	12.00000	16.00000	16.00000

U(1, 1)= 1.00000
U(1, 2)= 2.00000
U(1, 3)= 3.00000
U(1, 4)= 2.00000
U(1, 5)= 3.00000
U(2, 2)= 2.00000
U(2, 3)= 1.00000
U(2, 4)= 2.00000
U(2, 5)= 1.00000
U(3, 3)= 2.00000
U(3, 4)= 2.00000
U(3, 5)= 1.00000
U(4, 4)= 3.00000
U(4, 5)= 2.00000
U(5, 5)= 1.00000

MATRIX U

1.00000	2.00000	3.00000	2.00000	3.00000
0.0	2.00000	1.00000	2.00000	1.00000
0.0	0.0	2.00000	2.00000	1.00000
0.0	0.0	0.0	3.00000	2.00000
0.0	0.0	0.0	0.0	1.00000

CHAPTER III

FINITE DIFFERENCE ANALYSIS OF THE RESPONSE OF PORTAL FRAME

3.1 Program for Linear Equations of Motion of Multi-degree System

The basic equation governing the response on a multi-degree of freedom structure is

$$[M]\{\ddot{x}\} + [K]\{x\} = \{f(t)\} \tag{3.1}$$

where $[M]$ is the mass matrix, $[K]$ is the stiffness matrix, $\{\ddot{x}\}$ is the acceleration vector at any time t, $\{x\}$ is the displacement vector at time t and $\{f(t)\}$ is the disturbing force which varies with time. A vector iteration technique is utilized to determine the response vector $\{x(t)\}$ for the dynamic system. The first step is to rewrite Equation (3.1) in the new matrix form

$$\{\ddot{x}\} = [M]^{-1}\{f(t)\} - [M]^{-1}[K]\{x\} \tag{3.2}$$

where $[M]^{-1}$ is assumed to exist.

Recalling the Taylor series expansions of a function in one variable, it follows that

$$f(x) = \frac{f(x_0)}{0!} + \frac{(x_1-x_0)f'(x_0)}{1!} + \frac{(x-x_0)^2 f''(x_0)}{2!} + \ldots \tag{3.3}$$

Using a direct analogy the Tailor series expansion for a time-varying vector becomes

$$\{x_{(t)}\}^{(n+1)} = \{x_{(t)}\}^{n} + \Delta \{\dot{x}_{(t)}\}^{n} + \frac{\Delta^2}{2}\{\ddot{x}_{(t)}\}^{n} + \frac{\Delta^3}{6}\{\dddot{x}_{(t)}\}^{n} + \ldots \quad (3.4)$$

Differentiating Equation (3.4) with respect to time gives

$$\{\dot{x}\}^{(n+1)} = \{\dot{x}\}^{(n)} + \Delta \{\ddot{x}\}^{(n)} + \frac{\Delta^2}{2}\{\dddot{x}\}^{n} + \ldots \quad (3.5a)$$

$$\{\ddot{x}\}^{(n+1)} = \{\ddot{x}\}^{n} + \Delta \{\dddot{x}\}^{n} + \ldots \quad (3.5b)$$

where $\Delta = t^{(n+1)} - t^{n}$

The number of terms in this expausion may at first be arbitrarily chosen. The fewer the number of terms taken, the less accurate the result. The simplest solution may be found by considering no terms on the right hand sides of the expansions which contain derivatives higher than the second. Writing these in reverse order, gives

$$\{\ddot{x}\}^{n+1} = \{\ddot{x}\}^{n} \quad (3.6a)$$

$$\{\dot{x}\}^{n+1} = \{\dot{x}\}^{n} \quad (3.6b)$$

$$\{x\}^{n} = \{x\}^{n} + \Delta \{\dot{x}\}^{n} + \frac{\Delta^2}{2}\{\ddot{x}\}^{n} \quad (3.6c)$$

For a given value of $\{x\}^n$, $\{\dot{x}\}^n$ and $\{\ddot{x}\}^{n+1}$,

$\{\ddot{x}\}^n$ is found directly by use of Equation (3.6a); $\{\dot{x}\}^{n+1}$ is obtained from Equation (3.6b); $\{x\}^{n+1}$ is calculated from Equation (3.6c). Noting in Equation (3.6a) that the acceleration at the end of the interval is exactly the same as the acceleration at the beginning of the interval, one defines this procedure as the "constant acceleration method" of iteration (i.e. no derivatives beyond the second is retained). Permutting the value of n to (n - 1) in Equation (3.6a), (3.6b) and (3.6c) yields

$$\{x\}^n = \{x\}^{(n-1)} + \Delta\{\dot{x}\}^{(n-1)} + \frac{\Delta^2}{2}\{\ddot{x}\}^{(n-1)} \tag{3.7a}$$

$$\{\dot{x}\}^n = \{\dot{x}\}^{(n-1)} + \Delta\{\ddot{x}\}^{n-1} \tag{3.7b}$$

$$\{\ddot{x}\}^{(n-1)} = \frac{1}{\Delta}\{\dot{x}\}^n - \frac{1}{\Delta}\{\dot{x}\}^{(n-1)} \tag{3.7c}$$

By substituting $\{\ddot{x}\}^{(n-1)}$ in Equation (3.7c) into (3.7a) gives

$$\{x\}^n = \{x\}^{n-1} + \Delta\{\dot{x}\}^{(n-1)} + \frac{\Delta}{2}\{\dot{x}\}^n - \frac{\Delta}{2}\{\dot{x}\}^{(n-1)}$$

or

$$\{x\}^{n} = \{x\}^{(n-1)} + \frac{\Delta}{2}\{\dot{x}\}^{(n-1)} + \frac{\Delta}{2}\{\dot{x}\}^{n}$$

or

$$\{x\}^{n} - \{x\}^{(n-1)} - \frac{\Delta}{2}\left[\{\dot{x}\}^{(n-1)} + \{\dot{x}\}^{n}\right] = 0 \qquad (3.8)$$

Permutting the value of n to (n + 1) Equation (3.8), one obtains

$$\{x\}^{(n+1)} - \{x\}^{n} - \frac{\Delta}{2}\left[\{\dot{x}\}^{n} + \{\dot{x}\}^{(n+1)}\right] = 0 \qquad (3.9)$$

Subtracting Equation (3.8) from Equation (3.9) gives

$$\{x\}^{(n+1)} - 2\{x\}^{n} + \{x\}^{(n-1)} = \frac{\Delta}{2}\left[\{\dot{x}\}^{(n+1)} - \{\dot{x}\}^{(n-1)}\right] \qquad (3.10)$$

From Equation (3.7b) one obtains

$$\{\ddot{x}\}^{(n-1)} = \frac{1}{\Delta}\left[\{\dot{x}\}^{n} - \{\dot{x}\}^{(n-1)}\right] \qquad (3.11)$$

Combining Equations (3.10), (3.11) and the permuted form of Equation (3.6a) yields

$$\{\ddot{x}\}^{n} = \frac{1}{\Delta^2}\left[\{x\}^{(n+1)} - 2\{x\}^{n} + \{x^{(n-1)}\}\right] \qquad (3.12)$$

The convenient form of Equation (3.12) for calculation purposes is written as

$$\{x\}^{(n+1)} = 2\{x\}^{n} - \{x\}^{n-1} + \Delta t^2\{\ddot{x}\}^{n} \qquad (3.13)$$

Finally at $t = t_n$, it follows from Equation (3.2) that

$$\{\ddot{x}\}^{(n)} = -[M]^{-1}[K]\{x\}^{n} + [M]^{-1}\{f(t)\}^{n} \qquad (3.14)$$

From the above equations, the following steps are taken in the iteration procedure:

(1) At $t = 0$, $n = 0$, $\{x\}^{(0)} = 0$, $\{\ddot{x}\}^{(0)} = [M]^{-1}\{f\}^{(0)}$

and by choice $\{x\}^{(1)} = \frac{\Delta t^2}{2}[M]^{-1}\{f\}^{(0)}$

(2) At $t = t_1 = \Delta t$, $n = 1$

$$\{\ddot{x}\}^{(1)} = -[M]^{-1}[K]\{x\}^{(1)} + [M]^{-1}\{f\}^{(1)}$$

and

$$\{x\}^{(2)} = 2\{x\}^{(1)} - \{x\}^{(0)} + \Delta t^2\{\ddot{x}\}^{(1)}$$

(3) At $t = t_2 = 2\Delta t$, $n = 2$

$$\{\ddot{x}\}^{(2)} = -[M]^{-1}[K]\{x\}^{(2)} + [M]^{-1}\{f\}^{(2)}$$

$$\{x\}^{(3)} = 2\{x\}^{(2)} - \{x\}^{(1)} + \Delta t^2\{\ddot{x}\}^{(2)}$$

(4) At $t = t_i = i(\Delta t)$, $n = i$

$$\{\ddot{x}\}^{(i)} = -[M]^{-1}[K]\{x\}^{(i)} + [M]^{-1}\{f\}^{(i)}$$

$$\{x\}^{(i+1)} = 2\{x\}^{(i)} - \{x\}^{(i-1)} + \Delta t^2\{\ddot{x}\}^{(i)}$$

Additional steps similar to the latter steps is taken up to the value of n.

The Equation (3.1) may be applied to the solution for a single-bay multi-story frame shown in Figure 3.1

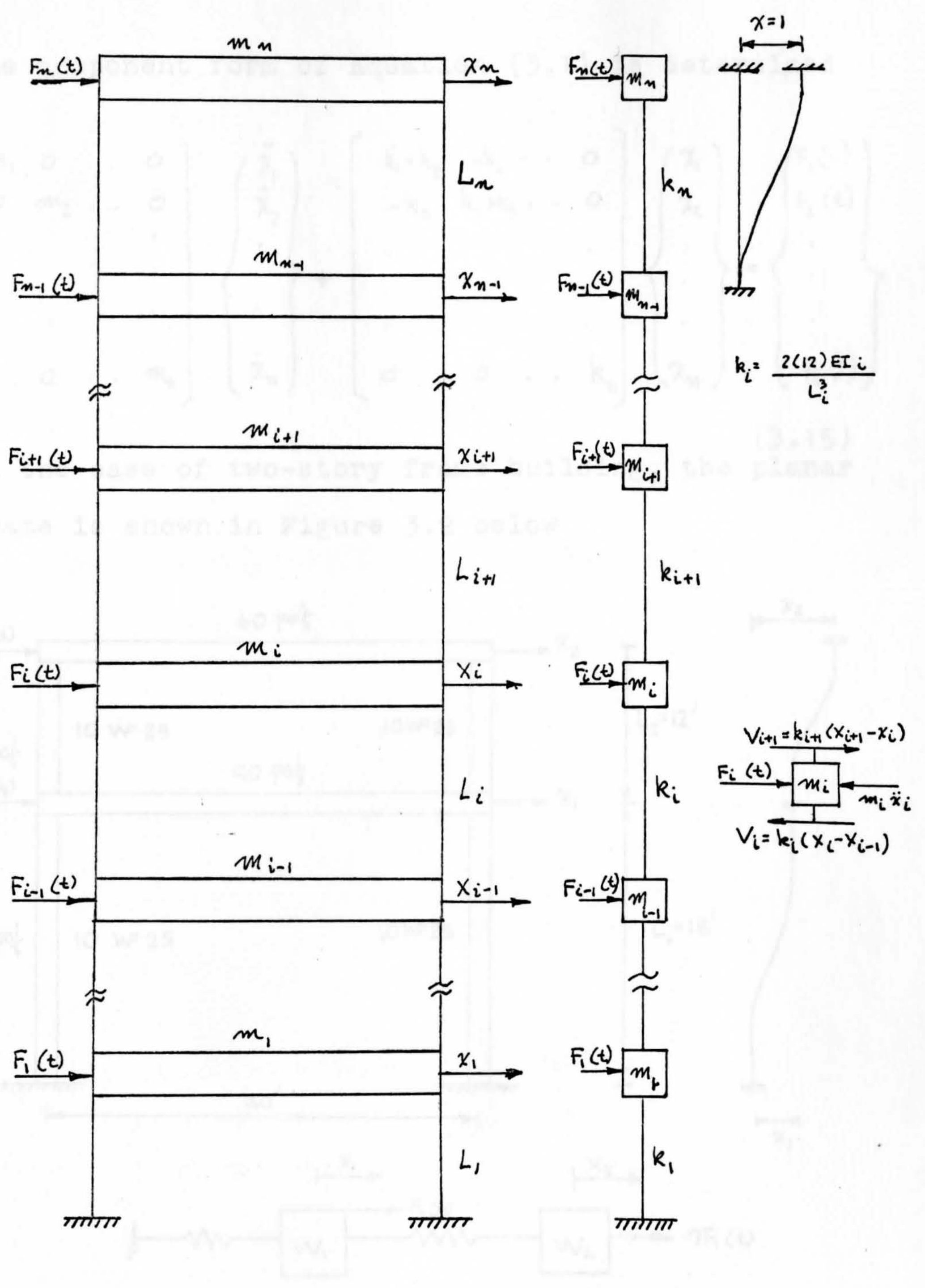

Figure 3.1 Multi-Story Portal Frame

The component form of Equation (3.1) is determined

$$\begin{bmatrix} m_1 & 0 & \cdots & 0 \\ 0 & m_2 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & m_n \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \\ \vdots \\ \ddot{x}_n \end{Bmatrix} + \begin{bmatrix} k_1+k_2 & -k_2 & \cdots & 0 \\ -k_2 & k_2+k_3 & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & k_n \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{Bmatrix} = \begin{Bmatrix} F_1(t) \\ F_2(t) \\ \vdots \\ F_n(t) \end{Bmatrix} \tag{3.15}$$

In the case of two-story frame building, the planar frame is shown in Figure 3.2 below

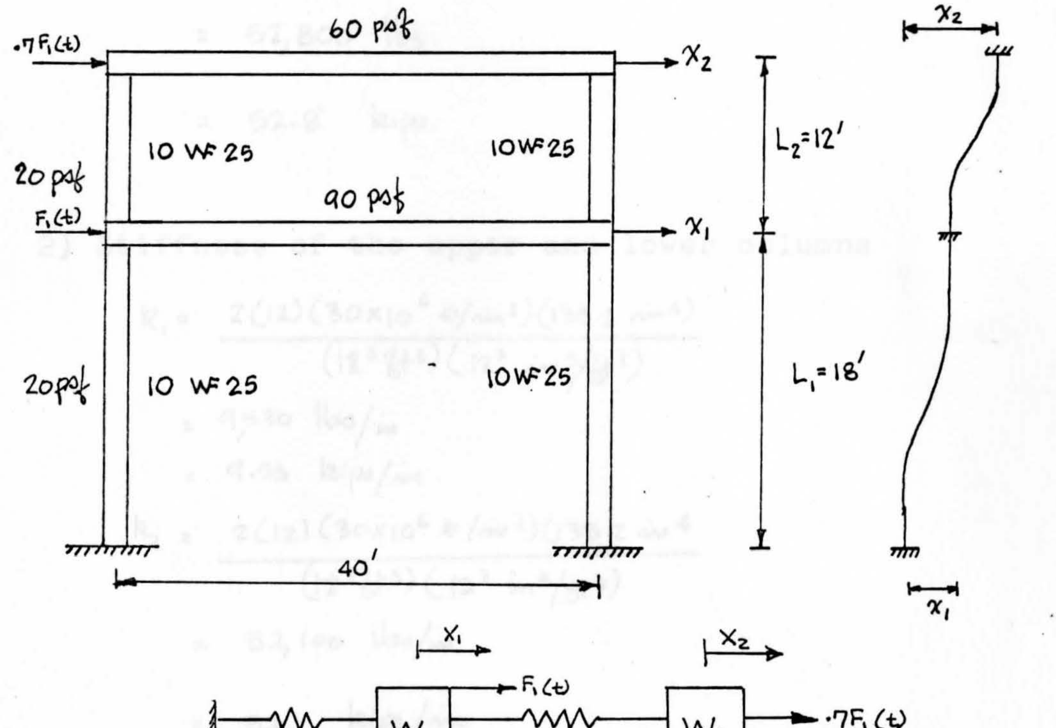

Figure 3.2 Two-Story Portal Frame

Equation (3.15) reduce to the form

$$\begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} k_1+k_2 & -k_2 \\ -k_2 & k_2 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} f_1(t) \\ f_2(t) \end{Bmatrix}$$

The numerical problem is analysed using the computer program with the following numerical input:

1) Weight of the upper and lower floors

$$W_1 = (90\ \#/ft^2)(40\ ft)(20\ ft) + (2)(20\ \#/ft^2)(15\ ft)(20\ ft)$$

$$= 84{,}000\ lbs$$

$$= 84.0\ kips.$$

$$W_2 = (60\ \#/ft^2)(40\ ft)(20\ ft) + (2)(20\ \#/ft^2)(6\ ft)(20\ ft)$$

$$= 52{,}800\ lbs$$

$$= 52.8\ kips.$$

2) Stiffness of the upper and lower columns

$$k_1 = \frac{2(12)(30\times10^6\ \#/in^2)(133.2\ in^4)}{(18^3\ ft^3)(12^3\ in^3/ft^3)}$$

$$= 9{,}530\ lbs/in$$

$$= 9.53\ kips/in$$

$$k_2 = \frac{2(12)(30\times10^6\ \#/in^2)(133.2\ in^4}{(12^3 ft^3)(12^3\ in^3/ft^3)}$$

$$= 32{,}100\ lbs/in$$

$$= 32.1\ kips/in$$

3) Mass of the upper and lower floors

$$m_1 = \frac{W_1}{g}$$

$$m_1 = \frac{84.0 \text{ kips}}{(32.2 \text{ ft/sec}^2)(12 \text{ in/ft})}$$

$$= 0.217 \text{ kips-sec}^2\text{/in}$$

$$m_2 = \frac{52.8 \text{ kips}}{(32.2 \text{ ft/sec}^2)(12 \text{ in/ft})}$$

$$= 0.137 \text{ kips-sec}^2\text{/in}$$

4) The natural frequencies and natural periods

$$f_1 = \frac{\omega_1}{2\pi} = 0.812 \text{ Hz}$$

$$f_2 = \frac{\omega_2}{2\pi} = 3.185 \text{ Hz}$$

then

$$T_1 = \frac{1}{f_1} = 1.232 \text{ sec.}$$

$$T_2 = \frac{1}{f_2} = 0.314 \text{ sec.}$$

5) Dynamic load function on the upper and lower floor

The time variation of the dynamic force F(t) is shown in Figure (3.3)

Figure 3.3 Time Variation of Dynamic force

$f_1=f(t)=10t+3.0$

$f_2=.7f(t) = -7t+2.7$

Equation (3.16) becomes

$$\begin{bmatrix} .217 & 0 \\ 0 & .137 \end{bmatrix} \begin{Bmatrix} \ddot{x}_1 \\ \ddot{x}_2 \end{Bmatrix} + \begin{bmatrix} 41.63 & -32.1 \\ -32.1 & 32.1 \end{bmatrix} \begin{Bmatrix} x_1 \\ x_2 \end{Bmatrix} = \begin{Bmatrix} f_1(t) \\ .7f_1(t) \end{Bmatrix}$$

The flow chart of the computer program for the linear equations of motion of the multi-degree of freedom system is shown in Figure (3.4) and the graph of the dynamic responses x_1 and x_2 are shown in Figure (3.5) and figure (3.6)

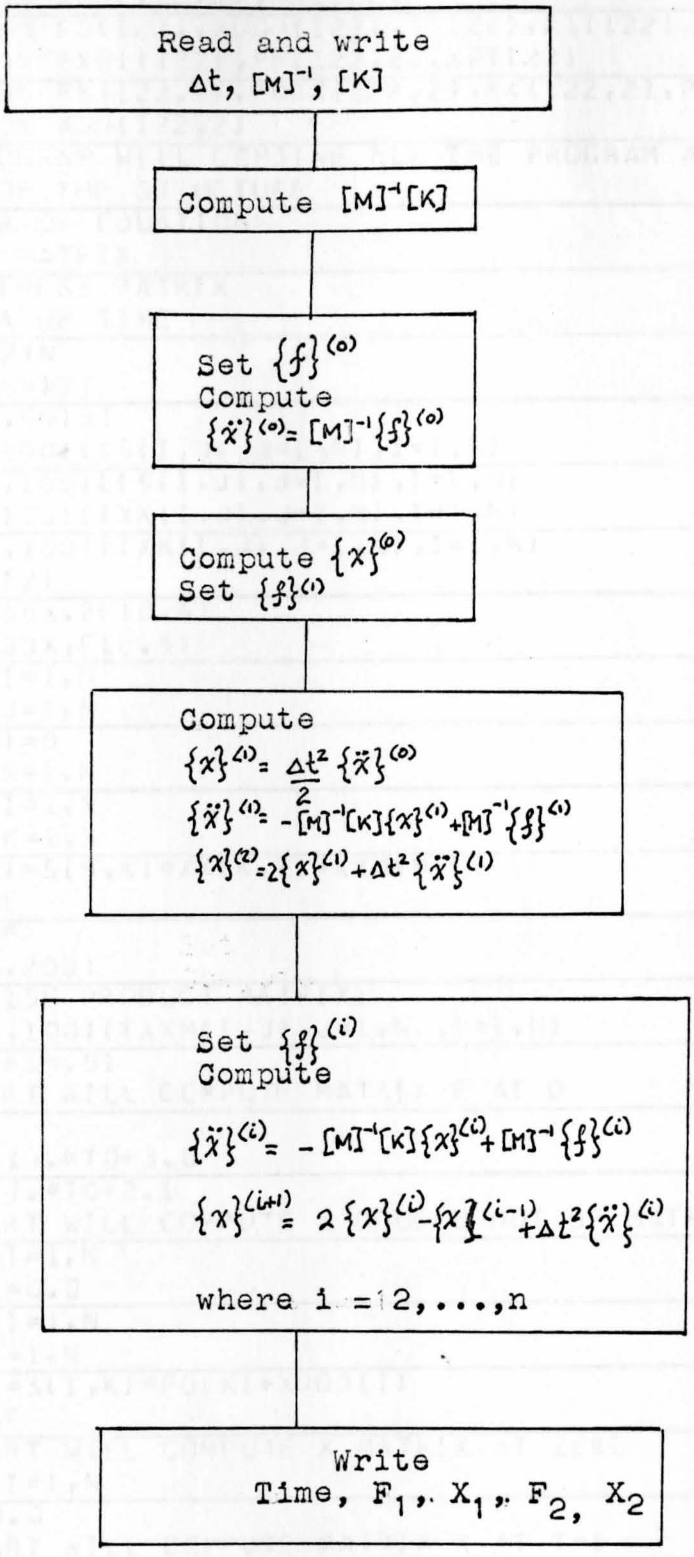

Figure 3.4 Flow chart of linear equations of motion.

```
      DIMENSION XK(122,2),S(72,72)
      DIMENSION XXM(122,2),FM(122)
      DIMENSION FO(122),XDDO(122),T(122),X1(122),F(122),PXDD1(122)
      DIMENSION PXD1(122),FF(122,2),XP(122)
      DIMENSION PX(122,2),PDDX(122,2),XX(122,2),PXDD(122,2),PXD(122,2)
      DIMENSION XDD(122,2)
C     THIS PROGRAM WILL COMBINE ALL THE PROGRAM AND GIVE THE FINAL
C     RESULT OF THE STRUCTURE
C     N=NUMBER OF EQUATIONS
C     XM=MASS MATRIX
C     XK=STIFFNESS MATRIX
C     DT=DELTA OF TIME
      READ(5,2)N
      READ(5,99)DT
      WRITE(6,99)DT
      READ(5,100)((S(I,J),J=1,N),I=1,N)
      WRITE(6,100)((S(I,J),J=1,N),I=1,N)
      READ(5,100)((XK(I,J),J=1,N),I=1,N)
      WRITE(6,100)((XK(I,J),J=1,N),I=1,N)
  2   FORMAT(I2)
  100 FORMAT(30X,2F10.4)
  99  FORMAT(30X,F10.5)
      DO 110 I=1,N
      DO 110 J=1,N
  110 XXM(I,J)=0
      DO 120 M=1,N
      DO 130 I=1,N
      DO 140 K=1,N
  140 XXM(M,I)=S(M,K)*XK(K,I)+XXM(M,I)
  130 CONTINUE
  120 CONTINUE
      WRITE(6,200)
  200 FORMAT(15H PRODUCT MATRIX)
      WRITE(6,100)((XXM(I,J),J=1,N),I=1,N)
  300 FORMAT(F15.5)
C     THIS PART WILL COMPUTE MATRIX F AT O
      TO=0.0
      FO(1)=-10.*TO+3.0
      FO(2)=-7.*TO+2.1
C     THIS PART WILL COMPUTE X DOUBLE DOT OF MATRIX X AT ZERO
      DO 180 I=1,N
  180 XDDO(I)=0.0
      DO 190 I=1,N
      DO 220K=1,N
  220 XDDO(I)=S(I,K)*FO(K)+XDDO(I)
  190 CONTINUE
C     THIS PART WILL COMPUTE X MATRIX AT ZERO
      DO 240 I=1,N
  240 XP(I)=0.0
C     THIS PART WILL COMPUTE MATRIX X AT T=1
      DO 260 I=1,N
  260 X1(I)=((DT**2.)/2.)*XDDO(I)
C     THIS PART WILL COMPUTE X DOUBLE DOT AT T=1
      F(1)=-10.*DT+3.0
      F(2)=-7.0*DT+2.1
      DO 270 M=1,N
      PXDD1(M)=0.0
  270 PXD1(M)=0.0
```

```
      DO 280 I=1,N
      DO 290 K=1,N
      PXDD1(I)=XXM(I,K)*X1(K)+PXDD1(I)
      PXD1(I)=S(I,K)*F(K)+PXD1(I)
  290 CONTINUE
  280 CONTINUE
      DO 301 I=1,N
      L=1
  301 XDD(L,I)=-PXDD1(I)+PXD1(I)
C     THIS PART WILL COMPUTE MATRIX X AT T=2
      M=2
      T(M)=2.*DT
      DO 310 I=1,N
      PX(L,I)=2.*X1(I)
      PDDX(L,I)=(DT**2.)*XDD(L,I)
      XX(M,I)=PX(L,I)+PDDX(L,I)
  310 CONTINUE
      L=2
  360 CONTINUE
      XX(1,1)=X1(1)
      XX(1,2)=X1(2)
      FF(M,1)=-10.*T(M)+3.0
      FF(M,2)=-7.*T(M)+2.10
      DO 315 I=1,N
      PXDD(L,I)=0.0
  315 PXD(L,I)=0.0
      DO 320 I=1,N
      DO 330 K=1,N
      PXDD(L,I)=-XXM(I,K)*XX(M,K)+PXDD(L,I)
      PXD(L,I)=S(I,K)*FF(M,I)+PXD(L,I)
  330 CONTINUE
  320 CONTINUE
      DO 340 I=1,N
      XDD(L,I)=PXDD(L,I)+PXD(L,I)
  340 CONTINUE
C     THIS PART WILL COMPUTE MATRIX X FOR ANY TIME
      LX=(.3/DT)+1
      LY=2*LX
      DO 350 I=1,N
      J=L+1
      JJ=L-1
      XX(J,I)=2.*XX(L,I)-XX(JJ,I)+DT**2.*XDD(L,I)
  350 WRITE(6,100) XX(J,I)
      M=M+1
      T(M)=M*DT
      L=L+1
  400 FORMAT(I3)
      IF(L-LX)360,360,370
  370 CONTINUE
      L=LX
  801 CONTINUE
      DO 803 I=1,N
  803 PXDD(L,I)=0.0
      DO 800 I=1,N
      DO 810 K=1,N
      PXDD(L,I)=-XXM(I,K)*XX(L,K)+PXDD(L,I)
  810 CONTINUE
  800 CONTINUE
```

```
      DO 820 I=1,N
      FF(L,I)=0.0
      J=L+1
      JJ=L-1
      XX(J,I)=2.*XX(L,I)-XX(JJ,I)+DT**2.*PXDD(L,I)
 820  CONTINUE
      L=L+1
      IF(L-LY)801,801,802
 802  CONTINUE
      MXO=0
      Y=0.0
      WRITE(6,500)
 500  FORMAT('1',7X,7H NUMBER,9X,5H TIME,12X,3H F1,8X,3H X1,8X,3H F2,5X,
     13H X2)
      WRITE(6,600)MXO,Y,FO(1),XP(1),FO(2),XP(2)
      MXO=1
      Y=DT
      WRITE(6,600)MXO,Y,F(1),X1(1),F(2),X1(2)
 600  FORMAT(9X,I3,10X,F10.5,5X,4F10.5)
      DO 700 I=2,LY
      Y=I*DT
      I1=1
      I2=2
 700  WRITE(6,600)I,Y,FF(I,I1),XX(I,I1),FF(I,I2),XX(I,I2)
      STOP
      END
```

VS LOADER
64

OPTIONS USED - PRINT,MAP,LET,CALL,NORES,NOTERM,SIZE=102400,NAME=**GO

NAME	TYPE	ADDR	NAME	TYPE	ADDR	NAME	TYPE	ADDR
MAIN	SD	120810	IHNECOMH*	SD	129DD8	IBCOM# *	LR	129E04
IHNCOMH2*	SD	12AD68	SEQDASD *	LR	12B196	IHNFRXPR*	SD	12B480
ADCON# *	LR	12B608	FCVAOUTP*	LR	12B6B2	FCVLOUTP*	LR	12B742
FCVEOUTP*	LR	12BD38	FCVCOUTP*	LR	12BE82	INT6SWCH*	LR	12C0F8
FIOCSBEP*	LR	12C17E	IHNFIOS2*	SD	12D170	IHNEFNTH*	SD	12D728
IHNUOPT *	SD	12DC70	IHNERRM *	SD	12DFD8	ERRMON *	LR	12DFD8
ALOG10 *	LR	12E5D8	ALOG *	LR	12E5F0	IHNSEXP *	SD	12E7B0
FQCONO# *	LR	12E960	IHNFCONI*	SD	12EE08	FQCONI# *	LR	12EE08
IHNTRCH *	LR	12F378	ERRTRA *	LR	12F380	IHNFTEN *	SD	12F620

TOTAL LENGTH EFA8
ENTRY ADDRESS 120810

0.02000

4.6082	0.0
0.0	7.2990
41.6300	-32.1000
-32.1000	32.1000

PRODUCT MATRIX

191.8394	-147.9232
-234.2979	234.2979

0.0232
0.0258
0.0400
0.0444
0.0603
0.0671
0.0837
0.0933
0.1095
0.1222
0.1371
0.1532
0.1658
0.1856
0.1950
0.2186
0.2240
0.2513
0.2521
0.2832
0.2788
0.3134
0.3033
0.3411
0.3252
0.3657
0.3437
0.3865
0.3583

NUMBER	TIME	F1	X1	F2	X2
0	0.0	3.00000	0.0	2.10000	0.0
1	0.00500	2.95000	0.00017	2.06500	0.00019
2	0.01000	2.90000	0.00069	2.03000	0.00076
3	0.01500	2.85000	0.00153	1.99500	0.00170
4	0.02000	2.80000	0.00270	1.96000	0.00300
5	0.02500	2.75000	0.00420	1.92500	0.00466
6	0.03000	2.70000	0.00601	1.89000	0.00666
7	0.03500	2.65000	0.00812	1.85500	0.00901
8	0.04000	2.60000	0.01054	1.82000	0.01169
9	0.04500	2.55000	0.01324	1.78500	0.01470
10	0.05000	2.50000	0.01623	1.75000	0.01802
11	0.05500	2.45000	0.01950	1.71500	0.02165
12	0.06000	2.40000	0.02304	1.68000	0.02558
13	0.06500	2.35000	0.02684	1.64500	0.02981
14	0.07000	2.30000	0.03089	1.61000	0.03431
15	0.07500	2.25000	0.03518	1.57500	0.03909
16	0.08000	2.20000	0.03971	1.54000	0.04414
17	0.08500	2.15000	0.04446	1.50500	0.04944
18	0.09000	2.10000	0.04944	1.47000	0.05498
19	0.09500	2.05000	0.05462	1.43500	0.06076
20	0.10000	2.00000	0.05999	1.40000	0.06677
21	0.10500	1.95000	0.06556	1.36500	0.07299
22	0.11000	1.90000	0.07131	1.33000	0.07942
23	0.11500	1.85000	0.07723	1.29500	0.08604
24	0.12000	1.80000	0.08331	1.26000	0.09285
25	0.12500	1.75000	0.08954	1.22500	0.09983
26	0.13000	1.70000	0.09591	1.19000	0.10698
27	0.13500	1.65000	0.10242	1.15500	0.11427
28	0.14000	1.60000	0.10904	1.12000	0.12171
29	0.14500	1.55000	0.11578	1.08500	0.12928
30	0.15000	1.50000	0.12262	1.05000	0.13697
31	0.15500	1.45000	0.12955	1.01500	0.14476
32	0.16000	1.40000	0.13656	0.98000	0.15265
33	0.16500	1.35000	0.14364	0.94500	0.16063
34	0.17000	1.30000	0.15078	0.91000	0.16868
35	0.17500	1.25000	0.15797	0.87500	0.17679
36	0.18000	1.20000	0.16520	0.84000	0.18495
37	0.18500	1.15000	0.17247	0.80500	0.19314
38	0.19000	1.10000	0.17975	0.77000	0.20137
39	0.19500	1.05000	0.18704	0.73500	0.20960
40	0.20000	1.00000	0.19433	0.70000	0.21784
41	0.20500	0.95000	0.20161	0.66500	0.22607
42	0.21000	0.90000	0.20887	0.63000	0.23428
43	0.21500	0.85000	0.21609	0.59500	0.24245
44	0.22000	0.80000	0.22327	0.56000	0.25058
45	0.22500	0.75000	0.23041	0.52500	0.25864
46	0.23000	0.70000	0.23748	0.49000	0.26664
47	0.23500	0.65000	0.24447	0.45500	0.27456
48	0.24000	0.60000	0.25139	0.42000	0.28239
49	0.24500	0.55000	0.25821	0.38500	0.29011
50	0.25000	0.50000	0.26493	0.35000	0.29771
51	0.25500	0.45000	0.27154	0.31500	0.30518
52	0.26000	0.40000	0.27803	0.28000	0.31252
53	0.26500	0.35000	0.28438	0.24500	0.31970
54	0.27000	0.30000	0.29060	0.21000	0.32672
55	0.27500	0.25000	0.29666	0.17500	0.33357
56	0.28000	0.20000	0.30256	0.14000	0.34024
57	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

6	0.31000	0.0	0.33409	0.0	0.37576
[illegible]	[illegible]	0.0	0.33863	0.0	0.38086
64	0.32000	0.0	0.34295	0.0	0.38571
65	0.32500	0.0	0.34706	0.0	0.39031
66	0.33000	0.0	0.35094	0.0	0.39466
67	0.33500	0.0	0.35460	0.0	0.39875
68	0.34000	0.0	0.35803	0.0	0.40258
69	0.34500	0.0	0.36124	0.0	0.40616
70	0.35000	0.0	0.36421	0.0	0.40946
71	0.35500	0.0	0.36695	0.0	0.41251
72	0.36000	0.0	0.36946	0.0	0.41528
73	0.36500	0.0	0.37173	0.0	0.41779
74	0.37000	0.0	0.37377	0.0	0.42003
75	0.37500	0.0	0.37556	0.0	0.42200
76	0.38000	0.0	0.37712	0.0	0.42369
77	0.38500	0.0	0.37843	0.0	0.42511
78	0.39000	0.0	0.37950	0.0	0.42626
79	0.39500	0.0	0.38032	0.0	0.42714
80	0.40000	0.0	0.38090	0.0	0.42774
81	0.40500	0.0	0.38124	0.0	0.42806
82	0.41000	0.0	0.38133	0.0	0.42812
83	0.41500	0.0	0.38117	0.0	0.42789
84	0.42000	0.0	0.38077	0.0	0.42740
85	0.42500	0.0	0.38013	0.0	0.42663
86	0.43000	0.0	0.37923	0.0	0.42559
87	0.43500	0.0	0.37810	0.0	0.42427
88	0.44000	0.0	0.37672	0.0	0.42269
89	0.44500	0.0	0.37509	0.0	0.42084
90	0.45000	0.0	0.37322	0.0	0.41872
91	0.45500	0.0	0.37111	0.0	0.41633
92	0.46000	0.0	0.36877	0.0	0.41368
93	0.46500	0.0	0.36618	0.0	0.41076
94	0.47000	0.0	0.36335	0.0	0.40759
95	0.47500	0.0	0.36029	0.0	0.40415
96	0.48000	0.0	0.35700	0.0	0.40046
97	0.48500	0.0	0.35347	0.0	0.39651
98	0.49000	0.0	0.34972	0.0	0.39231
99	0.49500	0.0	0.34574	0.0	0.38786
100	0.50000	0.0	0.34153	0.0	0.38317
101	0.50500	0.0	0.33711	0.0	0.37823
102	0.51000	0.0	0.33246	0.0	0.37305
103	0.51500	0.0	0.32760	0.0	0.36763
104	0.52000	0.0	0.32253	0.0	0.36198
105	0.52500	0.0	0.31725	0.0	0.35609
106	0.53000	0.0	0.31177	0.0	0.34998
107	0.53500	0.0	0.30609	0.0	0.34365
108	0.54000	0.0	0.30020	0.0	0.33709
109	0.54500	0.0	0.29413	0.0	0.33032
110	0.55000	0.0	0.28787	0.0	0.32334
111	0.55500	0.0	0.28142	0.0	0.31615
112	0.56000	0.0	0.27479	0.0	0.30875
113	0.56500	0.0	0.26798	0.0	0.30116
114	0.57000	0.0	0.26101	0.0	0.29337
115	0.57500	0.0	0.25386	0.0	0.28540
116	0.58000	0.0	0.24656	0.0	0.27723
117	0.58500	0.0	0.23909	0.0	0.26889
118	0.59000	0.0	0.23148	0.0	0.26038
119	0.59500	0.0	0.22372	0.0	0.25169
120	0.60000	0.0	0.21581	0.0	0.24284
121	0.60500	0.0	0.20777	0.0	0.23383
122	0.61000	0.0	0.19959	0.0	0.22467

NUMBER	TIME	F1	X1	F2	X2
0	0.0	3.00000	0.0	2.10000	0.0
1	0.01000	2.90000	0.00069	2.03000	0.00077
2	0.02000	2.80000	0.00272	1.96000	0.00301
3	0.03000	2.70000	0.00603	1.89000	0.00668
4	0.04000	2.60000	0.01056	1.82000	0.01172
5	0.05000	2.50000	0.01627	1.75000	0.01805
6	0.06000	2.40000	0.02308	1.68000	0.02562
7	0.07000	2.30000	0.03093	1.61000	0.03436
8	0.08000	2.20000	0.03976	1.54000	0.04419
9	0.09000	2.10000	0.04949	1.47000	0.05505
10	0.10000	2.00000	0.06006	1.40000	0.06684
11	0.11000	1.90000	0.07138	1.33000	0.07950
12	0.12000	1.80000	0.08339	1.26000	0.09294
13	0.13000	1.70000	0.09600	1.19000	0.10707
14	0.14000	1.60000	0.10914	1.12000	0.12182
15	0.15000	1.50000	0.12272	1.05000	0.13708
16	0.16000	1.40000	0.13666	0.98000	0.15277
17	0.17000	1.30000	0.15089	0.91000	0.16881
18	0.18000	1.20000	0.16533	0.84000	0.18508
19	0.19000	1.10000	0.17988	0.77000	0.20151
20	0.20000	1.00000	0.19446	0.70000	0.21799
21	0.21000	0.90000	0.20900	0.63000	0.23443
22	0.22000	0.80000	0.22342	0.56000	0.25074
23	0.23000	0.70000	0.23763	0.49000	0.26681
24	0.24000	0.60000	0.25154	0.42000	0.28256
25	0.25000	0.50000	0.26509	0.35000	0.29789
26	0.26000	0.40000	0.27819	0.28000	0.31270
27	0.27000	0.30000	0.29076	0.21000	0.32691
28	0.28000	0.20000	0.30273	0.14000	0.34043
29	0.29000	0.10000	0.31402	0.07000	0.35316
30	0.30000	0.0	0.32455	0.0	0.36503
31	0.31000	0.0	0.33426	0.0	0.37596
32	0.32000	0.0	0.34312	0.0	0.38590
33	0.33000	0.0	0.35110	0.0	0.39484
34	0.34000	0.0	0.35819	0.0	0.40276
35	0.35000	0.0	0.36436	0.0	0.40963
36	0.36000	0.0	0.36961	0.0	0.41544
37	0.37000	0.0	0.37391	0.0	0.42018
38	0.38000	0.0	0.37725	0.0	0.42383
39	0.39000	0.0	0.37962	0.0	0.42640
40	0.40000	0.0	0.38102	0.0	0.42786
41	0.41000	0.0	0.38144	0.0	0.42823
42	0.42000	0.0	0.38087	0.0	0.42750
43	0.43000	0.0	0.37932	0.0	0.42568
44	0.44000	0.0	0.37679	0.0	0.42278
45	0.45000	0.0	0.37329	0.0	0.41879
46	0.46000	0.0	0.36882	0.0	0.41374
47	0.47000	0.0	0.36340	0.0	0.40764
48	0.48000	0.0	0.35703	0.0	0.40050
49	0.49000	0.0	0.34974	0.0	0.39234
50	0.50000	0.0	0.34154	0.0	0.38319
51	0.51000	0.0	0.33246	0.0	0.37305
52	0.52000	0.0	0.32252	0.0	0.36197
53	0.53000	0.0	0.31175	0.0	0.34997
54	0.54000	0.0	0.30017	0.0	0.33706
55	0.55000	0.0	0.28782	0.0	0.32330
56	0.56000	0.0	0.27474	0.0	0.30870
57	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

NUMBER	TIME	F1	X1	F2	X2
0	0.0	3.00000	0.0	2.10000	0.0
1	0.02000	2.80000	0.00276	1.96000	0.00307
2	0.04000	2.60000	0.01066	1.82000	0.01183
3	0.06000	2.40000	0.02323	1.68000	0.02579
4	0.08000	2.20000	0.03997	1.54000	0.04442
5	0.10000	2.00000	0.06032	1.40000	0.06713
6	0.12000	1.80000	0.08370	1.26000	0.09328
7	0.14000	1.60000	0.10950	1.12000	0.12222
8	0.16000	1.40000	0.13708	0.98000	0.15324
9	0.18000	1.20000	0.16578	0.84000	0.18560
10	0.20000	1.00000	0.19496	0.70000	0.21856
11	0.22000	0.80000	0.22395	0.56000	0.25135
12	0.24000	0.60000	0.25210	0.42000	0.28320
13	0.26000	0.40000	0.27878	0.28000	0.31337
14	0.28000	0.20000	0.30333	0.14000	0.34111
15	0.30000	0.00000	0.32517	0.00000	0.36573
16	0.32000	0.0	0.34369	0.0	0.38654
17	0.34000	0.0	0.35871	0.0	0.40333
18	0.36000	0.0	0.37007	0.0	0.41594
19	0.38000	0.0	0.37764	0.0	0.42426
20	0.40000	0.0	0.38134	0.0	0.42820
21	0.42000	0.0	0.38111	0.0	0.42775
22	0.44000	0.0	0.37694	0.0	0.42293
23	0.46000	0.0	0.36888	0.0	0.41380
24	0.48000	0.0	0.35699	0.0	0.40046
25	0.50000	0.0	0.34141	0.0	0.38305
26	0.52000	0.0	0.32229	0.0	0.36173
27	0.54000	0.0	0.29984	0.0	0.33672
28	0.56000	0.0	0.27431	0.0	0.30825
29	0.58000	0.0	0.24597	0.0	0.27660
30	0.60000	0.0	0.21512	0.0	0.24208
31	0.62000	0.0	0.18208	0.0	0.20503
32	0.64000	0.0	0.14721	0.0	0.16583

NUMBER	TIME	F1	X1	F2	X2
0	0.0	3.00000	0.0	2.10000	0.0
1	0.03000	2.70000	0.00622	1.89000	0.00690
2	0.06000	2.40000	0.02348	1.68000	0.02607
3	0.09000	2.10000	0.05012	1.47000	0.05573
4	0.12000	1.80000	0.08423	1.26000	0.09386
5	0.15000	1.50000	0.12375	1.05000	0.13824
6	0.18000	1.20000	0.16654	0.84000	0.18646
7	0.21000	0.90000	0.21037	0.63000	0.23600
8	0.24000	0.60000	0.25304	0.42000	0.28427
9	0.27000	0.30000	0.29235	0.21000	0.32872
10	0.30000	0.0	0.32619	0.00000	0.36687
11	0.33000	0.0	0.35255	0.0	0.39645
12	0.36000	0.0	0.37083	0.0	0.41677
13	0.39000	0.0	0.38056	0.0	0.42740
14	0.42000	0.0	0.38149	0.0	0.42815
15	0.45000	0.0	0.37355	0.0	0.41907
16	0.48000	0.0	0.35691	0.0	0.40038
17	0.51000	0.0	0.33194	0.0	0.37253
18	0.54000	0.0	0.29927	0.0	0.33612
19	0.57000	0.0	0.25967	0.0	0.29194
20	0.60000	0.0	0.21410	0.0	0.24095
21	0.63000	0.0	0.16365	0.0	0.18430
22	0.66000	0.0	0.10947	0.0	0.12330

NUMBER	TIME	F1	X1	F2	X2
0	0.0	3.00000	0.0	2.10000	0.0
1	0.05000	2.50000	0.01728	1.75000	0.01916
2	0.10000	2.00000	0.06216	1.40000	0.06915
3	0.15000	1.50000	0.12584	1.05000	0.14060
4	0.20000	1.00000	0.19844	0.70000	0.22256
5	0.25000	0.50000	0.26970	0.35000	0.30317
6	0.30000	0.00000	0.32948	0.00000	0.37056
7	0.35000	0.0	0.36828	0.0	0.41389
8	0.40000	0.0	0.38351	0.0	0.43050
9	0.45000	0.0	0.37401	0.0	0.41959
10	0.50000	0.0	0.34031	0.0	0.38198
11	0.55000	0.0	0.28465	0.0	0.31997
12	0.60000	0.0	0.21080	0.0	0.23727
13	0.65000	0.0	0.12360	0.0	0.13906
14	0.70000	0.0	0.02854	0.0	0.03180

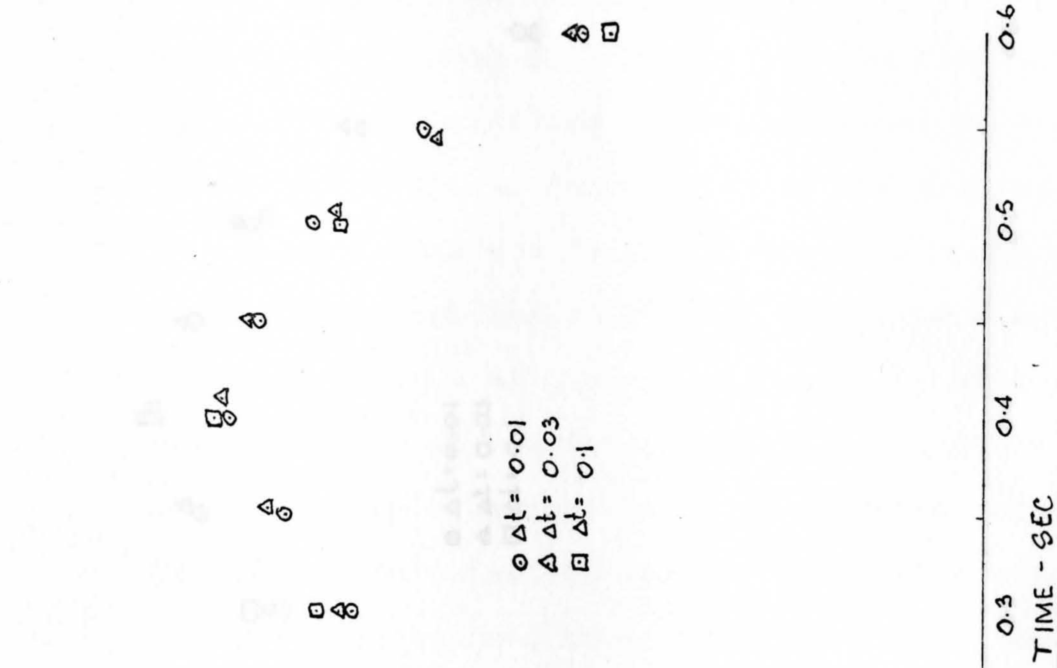

Figure 3.5 Comparible graph of displacement of X_1

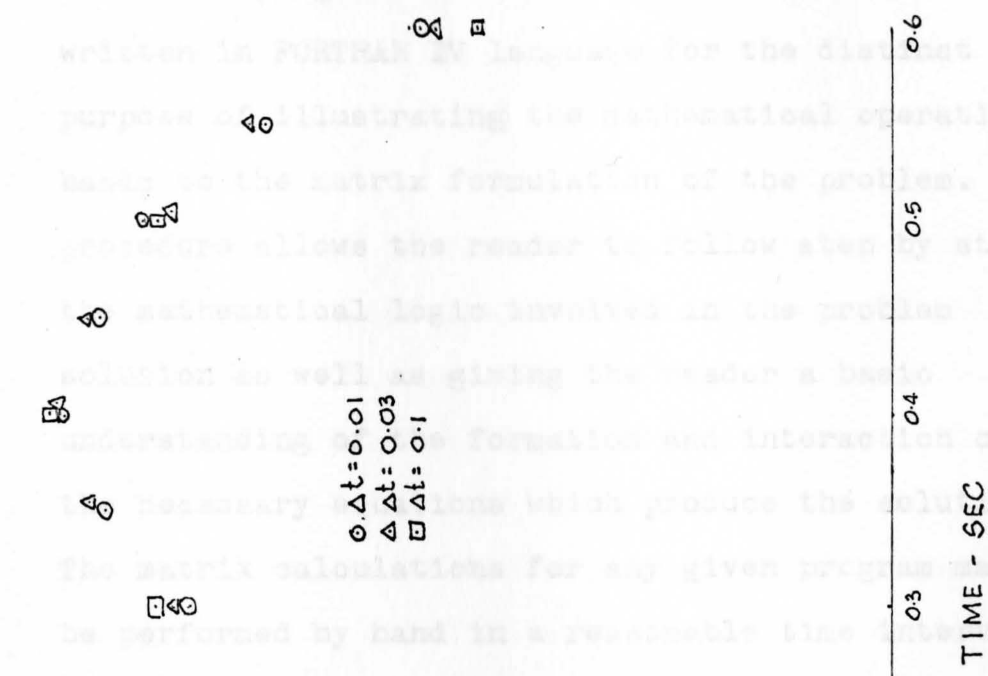

Figure 3.6 Comparible graph of displacement of X_2

CHAPTER IV

DISCUSSION AND CONCLUSION

4.1 Discussion

The programs in CHAPTER II are intentionally written in FORTRAN IV language for the distinct purpose of illustrating the mathematical operations basic to the matrix formulation of the problem. The procedure allows the reader to follow step by step the mathematical logic involved in the problem solution as well as giving the reader a basic understanding of the formation and interaction of the necessary equations which produce the solution. The matrix calculations for any given program may be performed by hand in a reasonable time interval for matrices of order three or less. If the order of the matrices is greater than three, the computer solutions offer the most efficient processes for problem solutions.

The FORTRAN programs developed in CHAPTER II which include the matrix determinant, matrix inversion, the characteristic equation determination, the evaluation of characteristic value and characteristic vectors, and the method of Cholesky transformation form a complete package of the usual matrix operations common to the field of Structural

Dynamics. In most cases, analyses of structures subject to dynamic loading usually involve a large number of degree of freedom which are efficiently processed by matrix techniques. Each of the above programs is specificly written to accommodate an arbitary number of degree of freedom and hence the computer package is useful for the range of problems encounted.

The program in CHAPTER III is also written in FORTRAN IV language. This program illustrates the usefulness of programming techniques to formulate the solution of a family of coupled linear differential equations utilizing finite difference methods. This technique replaces the rather complicated classical functional type solution with a simple numerical iterative method which strictly relies on a vast number of algebric operations for which the computer is extremely efficient in processing. The degree of accuracy using this method is bases upon only the size limitation of computer.

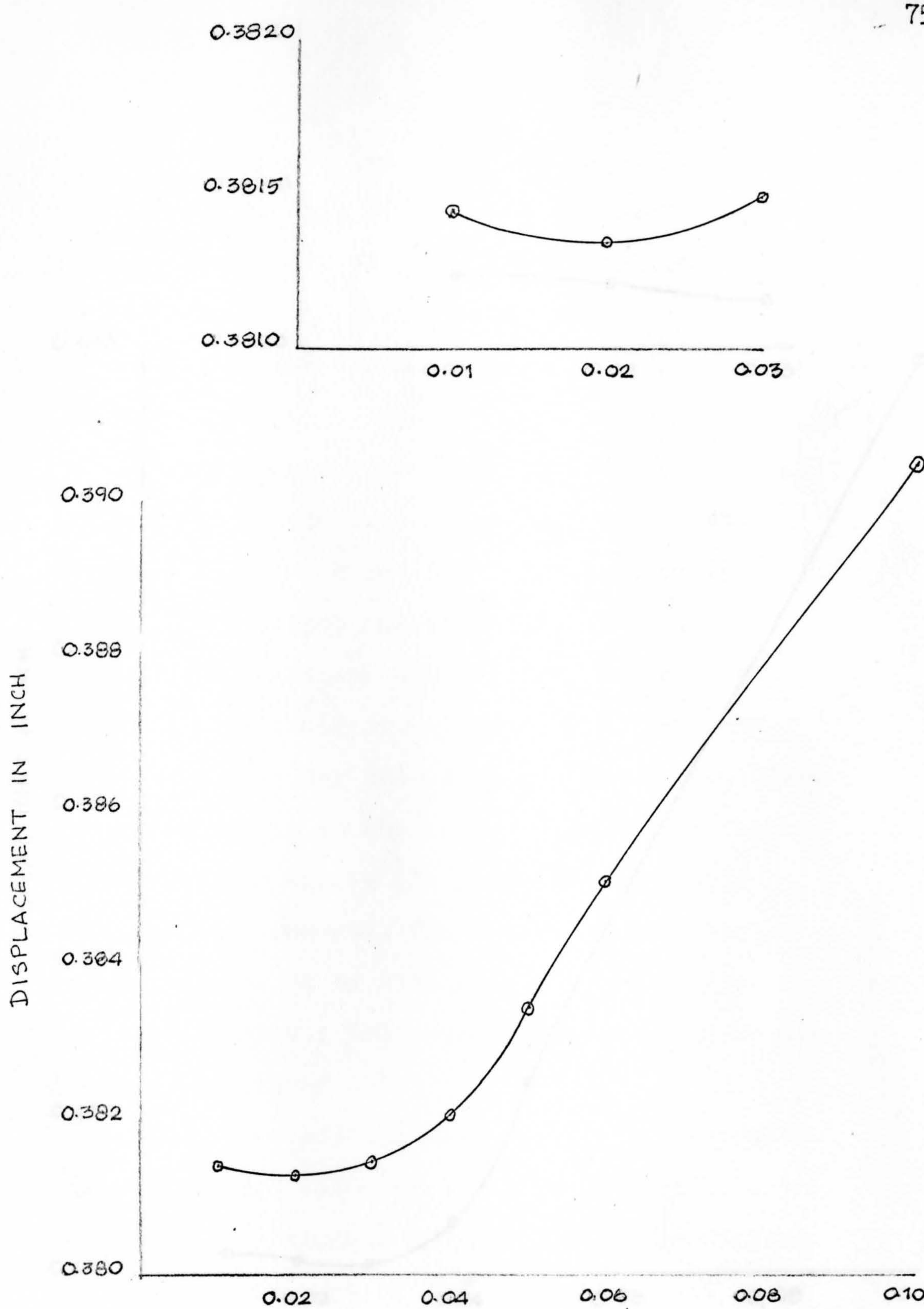

FIGURE 4.1 COMPARIBLE GRAPH OF MAXIMUM DISPLACEMENT OF Xi

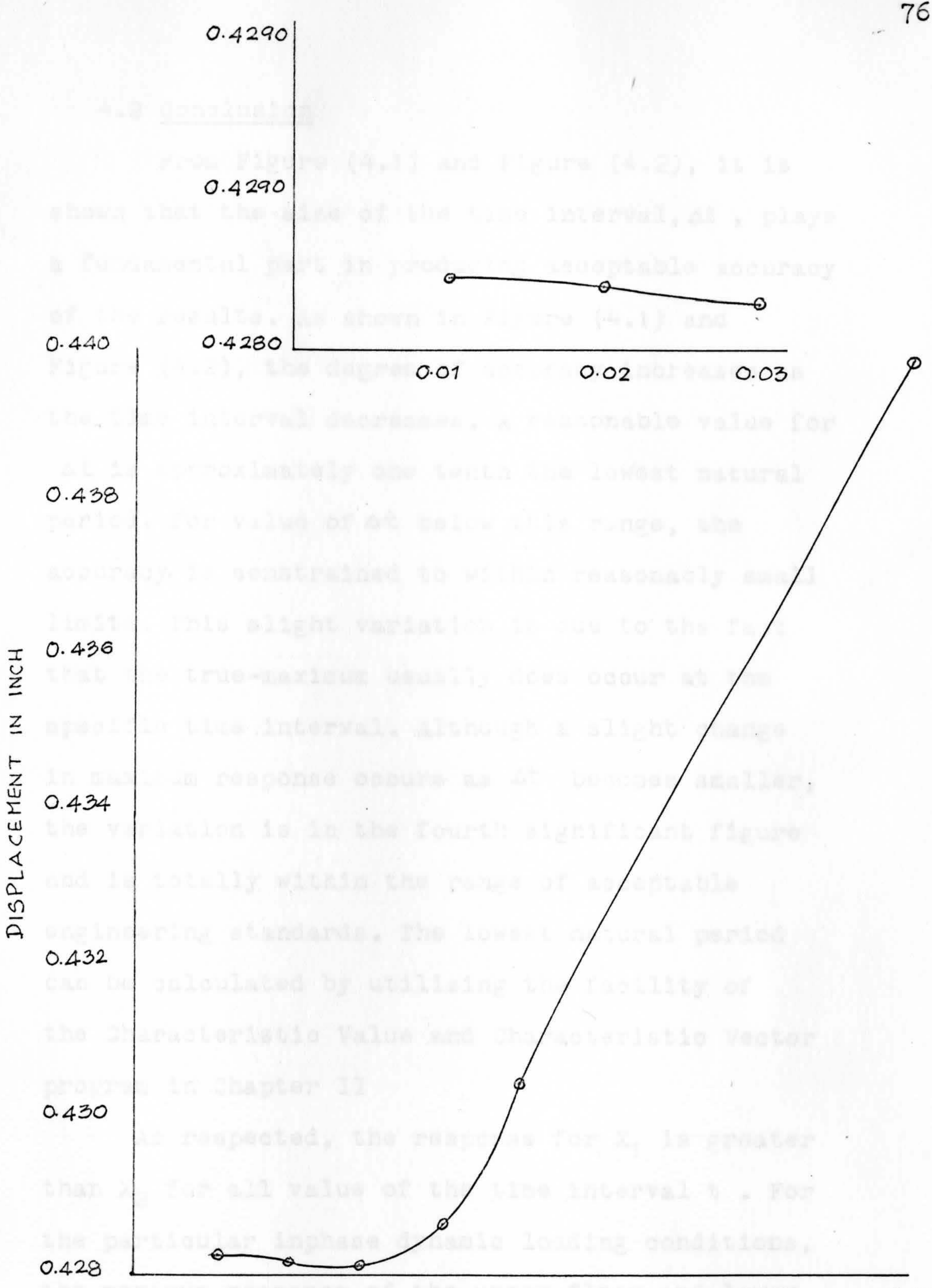

FIGURE 4.2 COMPARIBLE GRAPH OF MAXIMUM DISPLACEMENT OF X2

4.2 Conclusion

From Figure (4.1) and Figure (4.2), it is shown that the size of the time interval, Δt , plays a fundamental part in producing acceptable accuracy of the results. As shown in Figure (4.1) and Figure (4.2), the degree of accuracy increases as the time interval decreases. A reasonable value for Δt is approximately one tenth the lowest natural period. For value of Δt below this range, the accuracy is constrained to within reasonably small limits. This slight variation is due to the fact that the true-maximum usually does occur at the specific time interval. Although a slight change in maximum response occurs as Δt becomes smaller, the variation is in the fourth significant figure and is totally within the range of acceptable engineering standards. The lowest natural period can be calculated by utilizing the facility of the Characteristic Value and Characteristic Vector program in Chapter II

As respected, the response for X_1 is greater than X_2 for all value of the time interval t . For the particular inphase dynamic loading conditions, the maximum response of the upper floor and lower floor occurs at approximately the same time as shown in Figure (4.1) and Figure (4.2) which is an expected result.

For sample problem chosen, the maximum upper floor displacement is ≈0.38144 inch while the maximum lower floor displacement is ≈0.42823 inch. These values are within the acceptable elastic range of the material since the column lengths are 18 feet and 12 feet respectively and column size is a 10 WF 25.

BIBLIOGRAPHY

1. Pipes, L. A., "Matrix Mathods for Engineering", Prentice-Hall, Englewood Cliffs, N.J., 1963

2. Pipes, L. A. and Hovanessian S. A., "Matrix Computer Method in Engineering", John Wiley & Sons, Inc., New York, 1969

3. Westlake, J. R., "A Handbook of Numerical Matrix Inversion and Solution of Linear Equations", Robert E. Krieger Publishing Company, Huntington, 1975

4. Parsons, T. J., "Vibration of Lump-Mass Dynamical System using the Cholesky Transformation", Master Thesis, Youngstown State University, 1975

5. Rogers, G. L., "Dynamics of Framed Structures", John Wiley and Sons, Inc., New York, 1973

6. Fertis, D. G., "Dynamics and Vibration of Structures", John Wiley & Sons, New York, 1973